

**Statistical and Geometric Methods for  
Passive-blind Image Forensics**

Tian Tsong Ng

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

in the Graduate School of Arts and Sciences

Columbia University

2007

© 2007

Tian Tsong Ng

All Rights Reserved

ABSTRACT

## **Statistical and Geometric Methods for Passive-blind Image Forensics**

Tian Tsong Ng

Passive-blind image forensics (PBIF) refers to passive ways for evaluating image authenticity and detecting fake images. This dissertation proposes a physics-based approach for PBIF, with our definition of image authenticity derived from the image generative process comprising the 3D scene and the image acquisition device. We propose one statistical method and two geometric methods for capturing the image authenticity properties and addressing three separate problems in PBIF, i.e., detecting spliced images, distinguishing photographic images from photorealistic computer graphics, and estimating camera response function (CRF) from a single image. For image splicing detection, we show a statistical method for capturing the optical low-pass property of cameras. Through analysis on a proposed model of image splicing, we can explain the bicoherence response to image splicing better than the conventional quadratic phase coupling theory. Furthermore, we propose incorporating image-content-related features to improve the performance of image splicing detection. For distinguishing photographic images from photorealistic computer graphics, we propose a geometric method for capturing the properties of the object geometry, the object surface reflectance, and the CRF. The resulting geometry feature not only provides an intuitive understanding on how photographic images are different from photorealistic computer graphics, it also classifies the two types of images better than the wavelet characteristic feature and the features derived from modeling general computer graphics. For the work on CRF estimation,

we propose a geometric method based on geometric invariants for estimating CRF from a single-color-channel image. We provide an extensive analysis of the method and also propose a generalized gamma curve CRF model.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivations	1
1.2	Problem Formulation and Scope	3
1.3	Approach	6
1.3.1	Image Authenticity	6
1.3.2	Methods	6
1.4	Thesis overview	8
<b>2</b>	<b>A Statistical Method for Image Splicing Detection</b>	<b>11</b>
2.1	Introduction	11
2.2	Related Works	13
2.2.1	Audio Forgery Detection	13
2.2.2	Image Splicing Detection	13
2.3	Bicoherence	14
2.4	Bicoherence Theory for Splicing Detection	17
2.4.1	Splicing Model	17
2.5	Theory Validation	24
2.5.1	Columbia Image Splicing Detection Evaluation Dataset	24
2.5.2	Computation of Bicoherence Features	25

2.5.3	Experiments . . . . .	27
2.6	Improvement on Splicing Detection . . . . .	29
2.6.1	Edge Pixel Ratio Feature . . . . .	29
2.6.2	Image Structure Bicoherence Features . . . . .	31
2.7	Classification Experiment . . . . .	32
2.8	Discussions . . . . .	34
<b>3</b>	<b>A Geometric Method for Photographic and Computer Graphics</b>	
	<b>Images Classification</b>	<b>35</b>
3.1	Introduction . . . . .	35
3.2	Related Works . . . . .	37
3.3	Image Generation Process . . . . .	38
3.4	Geometry-based Image Description Framework . . . . .	40
3.4.1	Notation for This Chapter . . . . .	40
3.5	Linear Gaussian Scale-space . . . . .	41
3.6	Fractal Geometry . . . . .	42
3.7	Differential Geometry . . . . .	43
3.7.1	Gradient on Surface . . . . .	46
3.7.2	The Second Fundamental Form . . . . .	50
3.7.3	The Beltrami Flow Vector . . . . .	56
3.7.4	Normalization of Differential Geometry Features . . . . .	58
3.8	Local Patch Statistics . . . . .	60
3.9	Description of Joint Distribution by Rigid Body Moments . . . . .	62
3.10	Columbia Photographic Images and Photorealistic Computer Graphics Dataset . . . . .	65
3.11	Experiments . . . . .	69

3.12	An Online Demo System . . . . .	74
3.13	Discussion . . . . .	75
<b>4</b>	<b>A Geometric Method for Camera Response Function Estimation using a Single Image</b>	<b>77</b>
4.1	Introduction . . . . .	77
4.2	Prior Works on CRF Estimation . . . . .	80
4.3	Theoretical Aspects of the Algorithm . . . . .	81
4.3.1	Geometry Invariants . . . . .	81
4.3.2	General Properties of $\mathcal{G}_1$ . . . . .	82
4.3.3	Detection of Locally Planar Irradiance Points . . . . .	84
4.3.4	Geometric Significance of Equality Constraint . . . . .	87
4.3.5	CRF Estimation Model . . . . .	89
4.4	Addressing Detection Ambiguity Issues . . . . .	92
4.5	CRF Estimation . . . . .	97
4.5.1	Objective Function for CRF Estimation . . . . .	97
4.5.2	Joint Estimation for Multiple-channel Images . . . . .	98
4.6	Implementation Aspects of the Algorithm . . . . .	99
4.6.1	Computation of Image Derivative . . . . .	99
4.6.2	Error Metric Calibration . . . . .	100
4.6.3	Up-weighting Boundary Condition Data . . . . .	103
4.7	Experiments . . . . .	104
4.8	Limitation of the Proposed Method . . . . .	109
4.9	Discussion . . . . .	111
<b>5</b>	<b>Other Related Works</b>	<b>113</b>
5.1	Active Image Authentication: Digital Signatures and Watermarking . . . . .	113

5.2	Passive Image Authentication: Passive-blind Image Forensics (PBIF)	114
5.2.1	Image Forgery Detection	114
5.2.2	Image Source Identification	115
5.2.3	Image Operation Detection	116
5.2.4	Counter-attack Measure Design	117
<b>6</b>	<b>Conclusions</b>	<b>119</b>
6.1	Summary	119
6.2	Future Work	121
	<b>Appendix</b>	<b>125</b>
A	Proof for the Bipolar Effect on the Phase of the Spliced Signal Bicoherence Proposition (Proposition 1)	125
B	Proof for the Bipolar Effect on the Magnitude of the Spliced Signal Bicoherence Proposition (Proposition 2)	126
C	Derivation for Gradient on Surface	128
D	Online Demo System Implementation	129
D.1	Dataset with Non-photorealistic Computer Graphics	130
D.2	Image Downsizing	131
D.3	Classifier Fusion	132
D.4	Exploiting Dataset Heterogeneity	135
E	Proof of the Integral Solution to CRF Property (Property 5)	137
F	Proof of the Decomposition of $\mathcal{G}_1$ Proposition (Proposition 3)	138
G	Proof of the Geometric Significance of Equality Constraint Proposition (Proposition 4)	140
H	Proof of the Error Metric Calibration Proposition (Proposition 5)	141
I	The General Expression for Geometry Invariant Computation	144



## List of Figures

1.1	The front cover of the Feb 1982 National Geographic, showing two pyramids where one had been repositioned in order to fit the vertical cover frame. . . . .	2
1.2	Three examples of the well-known altered images appeared in the mainstream media and the Internet: (a) A composite of two original images appearing in Los Angeles Times in 2003, (b) A composite of two original images of John Kerry and Jane Fonda circulated in the Internet in 2004, (c) A 2006 Reuters image with the smoke being thickened and darkened using image editing software. . . . .	3
1.3	Two typical ways of creating fake images: (a) 2D image compositing, (b) 3D computer graphics rendering. . . . .	4
1.4	The relationship between the different PBIF areas of research according to our formulation. The works presented in this dissertation address a research problem in the areas highlighted in a yellow-shaded box. . . . .	5

1.5	The schematic image generative process, divided into the 3D scene process and the imaging device process, which respectively characterize the scene authenticity and the imaging-process authenticity. The imaging device process follows the model given in Tsing, Ramesh and Kanade [96]. Note that, the camera response function is the overall response of a camera, which represents the collective effect of all the processes within a camera. . . . .	7
1.6	A grayscale image can also be analyzed as a graph in a 3D space. Two different perspective views of a grayscale image graph are shown in this figure. . . . .	8
2.1	(a) The typical image forgery creation process. (b) An illustrative example for creating a composite image. . . . .	11
2.2	Two examples of spliced images without post-processing where each takes only 10-15 minutes to produce. . . . .	12
2.3	1D slices are extracted from a 2D image. A 1D slice is broken into overlapping segments for estimating bicoherence with Equ. 2.2. . . . .	15
2.4	(a) The model for an authentic and a spliced signal. (b) The bi-mode residue signal from the difference of an authentic signal and a spliced signal. . . . .	17
2.5	A bi-mode residue signal can be approximated as (a) a single dominant bipolar signal. (b) a sequence of bipolar signals. . . . .	18
2.6	A 3D plot for the numerator of the bipolar signal bicoherence for the case of $k = 1$ and $\Delta = 1$ . Note that the vertical axis is imaginary and therefore the resulting phase histogram has values concentrated at $\pm 90^\circ$ . . . . .	21

2.7	Example images from the <i>Columbia Image Splicing Detection Evaluation Dataset</i> . . . . .	24
2.8	The distribution of (a) the magnitude response $R_m$ and (b) the phase entropy $R_p$ for bicoherence computed on the Columbia Image Splicing Detection Evaluation Dataset. . . . .	27
2.9	(a) The average bicoherence phase histogram for the authentic image blocks and the spliced image blocks in the Columbia Image Splicing Detection Evaluation Dataset. (b) The difference histogram obtained by subtracting the authentic image block phase histogram from that of the spliced image blocks. . . . .	28
2.10	The distribution of the edge pixel ratio feature for the authentic and the spliced image blocks in our dataset. . . . .	29
2.11	The distribution of (a) the bicoherence magnitude response and (b) the bicoherence phase entropy for image blocks of the textured-textured, the textured-smooth, and the smooth-smooth categories in our dataset. . . . .	30
2.12	Examples for the structure-texture decomposition. . . . .	32
2.13	The distribution of (a) the bicoherence magnitude response $U_m$ and (b) the bicoherence phase entropy $U_p$ for the structure component of the authentic and the spliced image blocks. . . . .	33
2.14	The receiver operating characteristic (ROC) curve for the different feature combinations. . . . .	33
3.1	Examples of photo and PRCG from <a href="http://www.fakeorfoto.com">www.fakeorfoto.com</a> . . . . .	36
3.2	The geometry-based image description framework. . . . .	40
3.3	Log-log plot for estimating the fractal dimension of a $64 \times 64$ -pixel block from the tree (Left) and road (Right) region . . . . .	44

3.4	A typical concave camera response function. $M$ is the image irradiance function . . . . .	47
3.5	A visual effect of camera response function transform at compressing contrast at high intensity values and expanding contrast at the low intensity values. . . . .	47
3.6	(a) The relative strength of the modulation effect for different magnitude of the image irradiance gradient (b) The tail-compressing function, $S(x; 1)$ . . . . .	49
3.7	Distribution of the surface gradient for three different intensity ranges of the blue color channel. Each of the distribution is respectively computed from the entire image set . . . . .	51
3.8	(a) Illustration of the polygon effect. (b) Unusually sharp structure in computer graphics, note the pillar marked by the red ellipse. . . . .	51
3.9	(a) The typical shapes of the quadratic geometry (b) The shapes of the quadratic geometry in a 2D eigenvalue plot. Colors are for visual aid . . . . .	55
3.10	Distribution of the skewness of the 1st and 2nd eigenvalues of the second fundamental form for the blue color channel . . . . .	55
3.11	Comparing the joint distribution of the Beltrami flow components of a computer graphics image and that of its recaptured counterpart, the lines correspond to $y = x$ . . . . .	58
3.12	The grayscale patch (a) and the joint-spatial-color patch (b) are sampled at the edge points in an image. (c) Point masses on $S^7$ , a 7D sphere. . . . .	62
3.13	2D projection of the local patch feature distribution for the ( <i>Google+personal</i> image sets (red) and the <i>CG</i> image set (blue) . . . . .	65

3.14	2D projection of the fractal, the surface gradient, the 2nd fundamental form and the Beltrami flow vector features (from left to right) for the ( <i>Google+personal</i> image sets) (red) and the <i>CG</i> image set (blue)	65
3.15	Examples from our image sets. Note the photorealism of all images.	66
3.16	(a) Subcategories within <i>CG</i> and (b) Subcategories within <i>personal</i> image set, the number is the image count.	67
3.17	Receiver operating characteristic (ROC) curve for two classification experiments	70
3.18	Examples of the test image sets in Table 3.1	71
3.19	Classification performance of feature combinations (a) without and (b) with local fractal dimension features. Legend: g = surface gradient features, b = beltrami flow features, s = second fundamental form features, p = local patch statistics features.	73
3.20	The screen capture of the web demo user interface.	74
3.21	The image type for the user labels. The keyword for the image type is given in the bracket.	75
4.1	The computational steps for our CRF estimation method and their related implementation issues.	79
4.2	The gauge coordinates for computing $\mathcal{G}_1$ .	84
4.3	(a) Points detected by $E(R) < 10$ . (b) A magnified view showing the selected points being classifying into LPIP (blue) and non-LPIP (red), where LPIP is often surrounded by non-LPIP. (c) A local intensity profile of an LPIP	85

4.4	The CRF transformation preserves the shape of an isophote. A LPIP point with linear isophotes (as shown in the top row) in the irradiance domain retains the linear isophotes in the intensity domain and hence satisfies the equality constraint. It is possible that there exists a non-LPIP with a linear isophote (as shown in the bottom row) that satisfies the equality constraint. This results in a detection ambiguity in the point selection criterion using the equality constraint. . . . .	89
4.5	The typical $Q$ - $R$ histogram of LISO from single gamma-curve simulation images with $\gamma = 0.2, 0.4$ and $0.6$ for (a) without LPIP inference and (b) with LPIP inference. The red curve on the left is the marginal $Q$ distribution. The red line in each graph indicates the ground truth value of $\gamma$ . . . . .	92
4.6	The class-dependent feature distributions. . . . .	95
4.7	Relationship between LPIP posterior and the flatness measurement in an irradiance image. Note the log scale on the y-axis. . . . .	96
4.8	(a) The synthetic increasing-frequency sine function image, (b) the synthetic parabolic disk image. The red-color axis $x$ indicates the line along which the derivative profiles in Fig. 4.9 and Fig. 4.10 are extracted. . . . .	100

4.9	The derivation computation results for the synthetic increasing-frequency sine function image in Fig. 4.8 (a). The results shown are the profiles extracted along the red-color axis $x$ in Fig. 4.8 (a). From the top row to the bottom row are the estimation results of the function $R$ , its first-order derivative $R_x$ , and its second-order derivative $R_{xx}$ . From the left-most column to the right-most column are the estimation results computed by the smoothing cubic B-spline, local 3rd-order polynomial fitting with a $17 \times 17$ kernel, and the local 3rd-order polynomial fitting with a $7 \times 7$ kernel. . . . .	101
4.10	The derivation computation results for the synthetic parabolic disk function image in Fig. 4.8 (b). The results shown are the profiles extracted along the red-color axis $x$ in Fig. 4.8 (b). From the top row to the bottom row are the estimation results of the function $R$ , its first-order derivative $R_x$ , and its second-order derivative $R_{xx}$ . From the left-most column to the right-most column are the estimation results computed by the smoothing cubic B-spline, local 3rd-order polynomial fitting with a $17 \times 17$ kernel, and the local 3rd-order polynomial fitting with a $7 \times 7$ kernel. . . . .	102
4.11	A plot of RMSE mean (Left) and RMSE 2nd-moment (Right) for different cameras and different CRF estimation strategies $E_1$ to $E_4$ and $E_{rgb}$ . . . . .	104
4.12	Estimated blue-color channel CRF's for the five models of camera using a single color-channel image ( $E_1$ ). The thick blue line represents the ground-truth CRF. The CRF of Canon RebelXT and Nikon D70 are most different and the estimated CRF for Canon RebelXT and Nikon D70 are shown in the lower right subplot. . . . .	106

4.13	Estimated blue-color channel CRF's for the five models of camera using four blue-color-channel images ( $E_4$ ). The thick blue line represents the ground-truth CRF. The CRF of Canon RebelXT and Nikon D70 are most different and the estimated CRF for Canon RebelXT and Nikon D70 are shown in the lower right subplot. . . . .	107
4.14	Estimated blue-color channel CRF's of Canon RebelXT and Nikon D70 for $E_1$ , $E_{rgb}$ , $E_2$ , $E_3$ and $E_4$ . The thick blue line represents the ground-truth CRF. . . . .	108
4.15	Curve-fitting in $\bar{Q} \times R$ space with data of (a) high R-coverage, (b) Low R-coverage. The thick blue line represents the ground-truth $Q(R)$ curve. . . . .	108
4.16	The distribution of $(R, Q)$ points computed from an irradiance image (i.e., a gamma transformed image with $\gamma = 1$ ). The distribution is represented by a 2D $R$ - $Q$ histogram obtained from counting the point weight assigned through the LPIP inference. Figure (a) shows the result of the LPIP inference trained using gamma images of $\gamma = 0.2, 0.4, \text{ and } 0.6$ . Figure (b) shows that from the same training set but with images of $\gamma = 1.0$ included. . . . .	111
4.17	Example images that the CRF estimation algorithm perform less well.	112
6.1	The two main approaches for PBIF: the statistical approach and the physics-based approach. . . . .	120
6.2	An illustrative example of checking the consistency of the scene illumination extracted from the eye image of two different human subjects in the same image. . . . .	123



6.3	An illustrative example of performing PBIF using more than a single image. . . . .	124
D.1	Example images from the four categories of the online classifier training dataset. . . . .	130
D.2	Nine combinations of the data subsets. . . . .	136

## List of Tables

2.1	SVM Classification Accuracy for Different Feature Combinations . . .	34
3.1	Classifier Test Accuracy . . . . .	70
4.1	Mean RMSE ( $\times 10^{-2}$ ) of the proposed CRF model . . . . .	91
4.2	Overall RMSE ( $\times 10^{-2}$ ) for CRF Estimation . . . . .	104
D.1	SVM Classification Accuracy for Different Image Size Reduction Strategies . . . . .	132
D.2	Downsized Image Classification Accuracy for Different Fusion Strategies	134
D.3	Classification Accuracy After Considering Dataset Heterogeneity . . .	136

## Acknowledgements

This dissertation is the fruit of many people who have left their imprints on me, at present and in the past. I would like to thank my previous thesis advisor, Stephen Lu and Nick Kingsbury. I would like to express my sincere gratitude to my present thesis advisor, Professor Shih-Fu Chang, who guided me into a right direction with generosity and helped me see my own perpetual insufficiency.

I would like to thank Mao-Pei Tsui, who has encouraged and inspired me whenever there were seemingly unsurmountable difficulties. Thanks also to all my colleagues, especially Lexing Xie, Jessie Hsu, Dongqing Zhang, and Jun Wang who constantly reminded me that I am never alone.

I would like to thank Shree Nayar and the member of his lab, especially Li Zhang, Kshitiz Garg, Gurunandan Krishnan, Jinwei Gu, who generously granted me access to their lab equipment and taught me knowledge and experimental skills in computer vision.

I would like to thank Peter Belhumeur, Shree Nayar, Tony Jebara, Richard Hamilton, Ioannis Karatzas, and Duong Hong Phong whose lectures inspired my work and kept my academic interests alive. I would like to thank all the members of my defense committee, Peter Belhumeur, Dan Ellis, Shree Nayar, Xiaodong Wang, who ungrudgingly sacrificed their time and effort to give the finishing touch to this dissertation.

I would like to thank Micheal Lim and Eric Zavesky for proofreading this dissertation. I would like to thank A\*STAR of Singapore for supporting my studies in Columbia University.

Finally, I would like to thank my parents for seeing me through a life here not without difficulties in this land away from home.

# Chapter 1

## Introduction

### 1.1 Motivations

In February 1982, an altered image of the Great Pyramids in Giza appeared on the front cover of the National Geographic magazine (see Figure 1.1). One of two pyramids in the image was repositioned and moved closer to the other so that they both fitted well into the vertical front cover. This was one of the earliest incidents that signified the degree of manipulation enabled by image editing software, and had an impact that caused ‘The earth moved in the world of photography’ [82], as it challenged our conventional belief of ‘seeing is believing’. Since then, as the world is getting more digital, image manipulation has become easier and more versatile than ever, with the advances in computers and image editing tools such as Adobe Photoshop. Nowadays, image alteration in the mainstream media has become common. In 1989, *The Wall Street Journal* estimated that 10% of all color photographs published in United States were digitally altered or retouched [2]. Three recent and well-known altered images are shown in Figure 1.2. Apart from the mainstream media, the Internet hosts an enormous number of images, where many of them have no provenance information or certainty of authenticity. For example, a website

Cover of National Geographic, Feb 1982

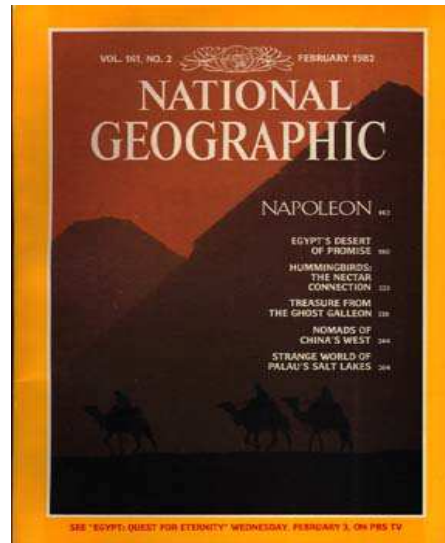


image source: <http://commfaculty.fullerton.edu/lester/writings/geo.html>

Figure 1.1: The front cover of the Feb 1982 National Geographic, showing two pyramids where one had been repositioned in order to fit the vertical cover frame.

[www.worth1000.com](http://www.worth1000.com), as of December 2006, hosts as many as 273,500 photorealistic images created using Adobe Photoshop. Therefore, it is crucial to develop a scientific and automatic way for assessing image authenticity, which is the theme of this dissertation - *passive-blind image forensics* (PBIF). Such image forensic techniques have a wide range of applications in news reporting, journalism, legal services, intelligence services, forensics investigation, insurance claim investigation, financial systems, and e-commerce. The importance for PBIF will continue to grow in the years to come, as the world enters the third wave of globalization [29] with the rise of the individual-centric media, typified by the culture of blogging and media sharing. The individual-centric media will generate an unprecedented number of newsworthy images, where their authenticity needs to be ascertained.

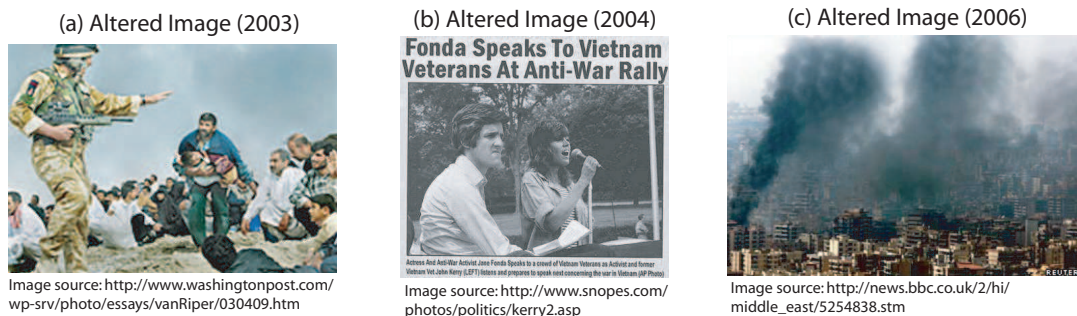


Figure 1.2: Three examples of the well-known altered images appeared in the mainstream media and the Internet: (a) A composite of two original images appearing in Los Angeles Times in 2003, (b) A composite of two original images of John Kerry and Jane Fonda circulated in the Internet in 2004, (c) A 2006 Reuters image with the smoke being thickened and darkened using image editing software.

## 1.2 Problem Formulation and Scope

In contrast to an authentic image captured by a camera, a fake or tampered image is one that bears falsehood in event, place, or time. The goal of passive-blind image forensics is to detect fake images or more generally evaluate the authenticity of an image. The two major methods for creating fake images are 2D image compositing and 3D computer graphics rendering (see Fig. 1.3). In this dissertation, we identify the detection of the 2D composite images as image forgery detection and the detection of 3D computer graphic synthesized images as image source identification.

Another recent thread of development in image synthesis is image-based rendering [86], where a novel-view image is synthesized from a set photographic images, or a photorealistic image is obtained when a computer graphic object is given the appearance of a real object which is extracted from photographs of multiple views and lightings. In essence, images synthesized by image-based rendering are the hybrid of photographic images and computer graphics rendering. Since such tampering methods are usually complex, non-professional attackers will not be able to access such tools. Therefore, we focus on detection of 2D composite images and computer

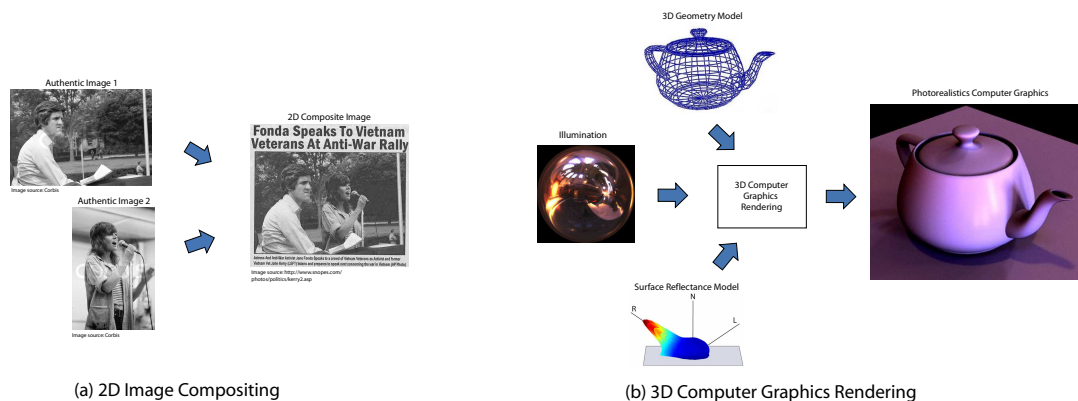


Figure 1.3: Two typical ways of creating fake images: (a) 2D image compositing, (b) 3D computer graphics rendering.

graphics images in this thesis. Besides image-based rendering, there is also another class of image synthesis techniques used for filling in the removed regions within an images, by texture synthesis [16] or image inpainting [5, 6, 75]. However, these techniques are often restricted either to the synthesis of regular textures or filling in small-area regions. The consideration of such tampering techniques is also beyond the scope of this thesis.

For image source identification, one can define image sources at different levels of specificity. At the coarse level, one can in general collectively define camera images as having a single source and 3D computer graphic images as having a different source. At the intermediate level, one can define images captured by different models of camera, generated by different models of printers or scanners, or rendering by different 3D computer graphic rendering techniques respectively as having different sources. Whereas at the finest level, one can define images as captured by different cameras (even different units of the same model), or generated by different printer or scanners. Capability for distinguishing image sources of finer specificity are useful for image forgery detection as the image fragments forming a composite image are likely to have different fine sources (e.g., from different models of camera).

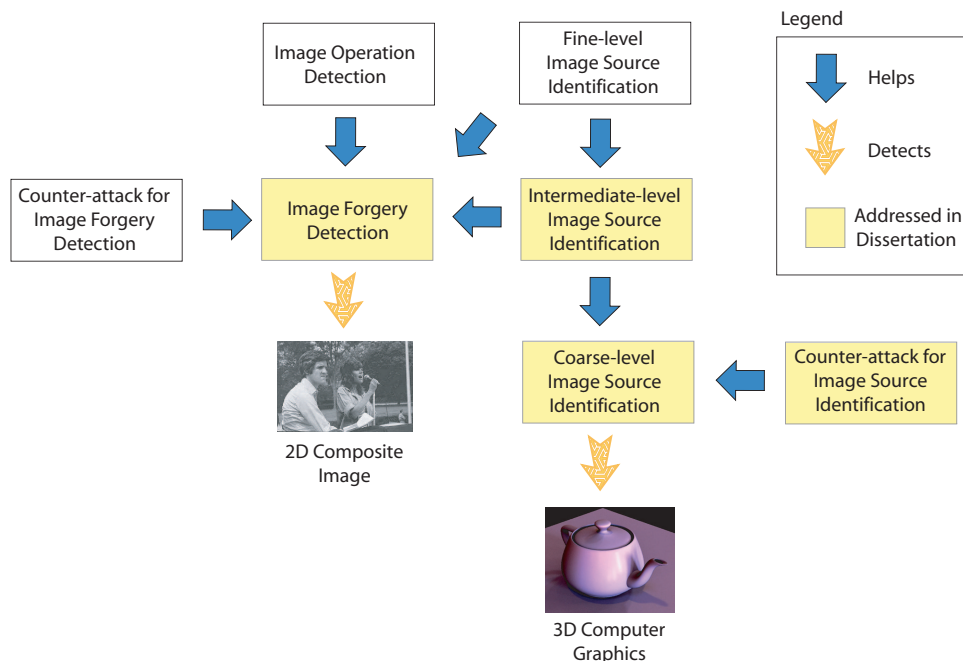


Figure 1.4: The relationship between the different PBIF areas of research according to our formulation. The works presented in this dissertation address a research problem in the areas highlighted in a yellow-shaded box.

Figure 1.4 shows the relationship between the different areas of research in PBIF according to our formulation. Apart from the main areas of image forgery detection and image source identification, the other auxiliary PBIF areas of research are image operation detection and counter-attack measure design. As composite image creation often involves post-processing such as image smoothing and noise addition, image operation detection indirectly helps image forgery detection. As fake image creators could *attack* a PBIF system by specifically processing the fake images such that they escape detection, a critical issue in PBIF is to come up with an effective counter-attack measure to make a PBIF system robust to the forger’s attack. The works presented in this dissertation address a problem in the areas of research highlighted in a yellow-shaded box in Figure 1.4.



## 1.3 Approach

### 1.3.1 Image Authenticity

An image is authentic if it represents a witness to an actual event, place, or time. Image authenticity is a central idea for addressing the PBIF problems. A definition of image authenticity should enable us to distinguish an authentic image from the fake images, such as the 2D composite images and the 3D computer graphics images. We define image authenticity as the characteristics of an image generative process, for which we divide into the 3D scene process and the imaging device process, as shown in Fig. 1.5. We name the image authenticity quality associated to the 3D scene process as the *scene authenticity* and that of the imaging device process as the *imaging-process authenticity*. Scene authenticity is governed by the physics of light transport in a 3D scene, while imaging-process authenticity is governed by the characteristics of the sequence of operations within an imaging device (the operations explained in Subsec. 2.2.2). The work presented in this dissertation captures the specific authenticity properties shown in the yellow-shaded boxes in Fig. 1.5.

### 1.3.2 Methods

We approach the PBIF problems through statistical methods and geometric methods. For a statistical method, an image is treated as a random signal, from which image-authenticity related statistical quantities can be extracted. Whereas for a geometric method, an image is treated as a graph of its intensity function as shown in Fig. 1.6 (see Chapter 2). Such a graph qualifies as a submanifold embedded in the Eculidean space, from which image-authenticity related geometric quantities can be extracted (see Chapter 3 and Chapter 4).

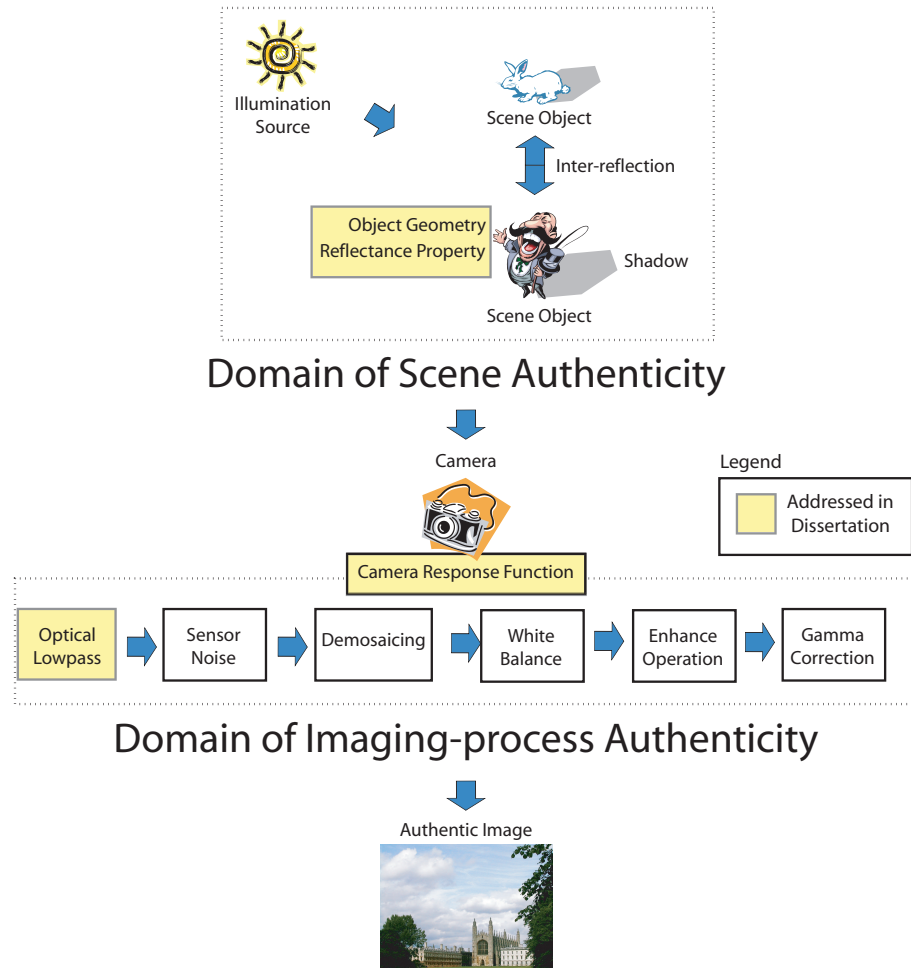


Figure 1.5: The schematic image generative process, divided into the 3D scene process and the imaging device process, which respectively characterize the scene authenticity and the imaging-process authenticity. The imaging device process follows the model given in Tsin, Ramesh and Kanade [96]. Note that, the camera response function is the overall response of a camera, which represents the collective effect of all the processes within a camera.

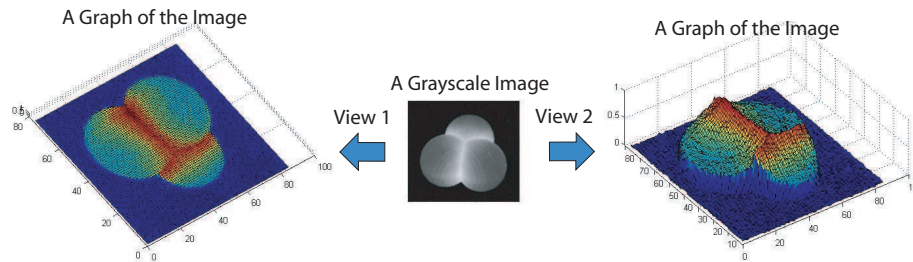


Figure 1.6: A grayscale image can also be analyzed as a graph in a 3D space. Two different perspective views of a grayscale image graph are shown in this figure.

## 1.4 Thesis overview

In this dissertation, we present three works which exemplify the statistical and the geometric approaches. In Chapter 2, we present a statistical method for image splicing detection, which is a fundamental problem in image forgery detection. The statistical method uses bicoherence, a third-order normalized moment spectral, to characterize the camera optical low-pass property, which is part of the imaging-process authenticity. In this work, we provide a theoretical analysis which connects the magnitude and the phase response of the image bicoherence to the camera optical low-pass property. This theoretical results justify the use of the bicoherence magnitude and phase features for image splicing detection. As an image originally contains a significant level of bicoherence energy, it makes the detection of the splicing response in bicoherence difficult. To overcome this baseline noise problem, we extract image content features from image edges and the structure component of an image (without the fine image texture). In this work, we created the *Columbia Image Splicing Detection Evaluation Dataset* [66] which consists of 933 authentic and 912 spliced grayscale image blocks of size  $128 \times 128$  pixels. This dataset is open for research purposes.

In Chapter 3, we present a geometric method for distinguishing photographic im-

ages and photorealistic computer graphics images, which is a problem of coarse-level image source identification. The geometric method captures the differences between the photographic image generative process and the photorealistic computer graphics generative process, in terms of the scene object geometry, the scene object reflectance property, and the camera response function. The first two properties belong to the scene authenticity, while the third one belongs to the imaging-process authenticity. The geometric method captures these properties with a set of differential-geometric quantities. Apart from these quantities, we also capture the difference between these two types of images using local fractal dimension and local patch statistics. In this work, we also created the *Columbia Photographic Images and Photorealistic Computer Graphics Dataset* [69] which consists of two sets of 800 photographic images, one set of 800 photorealistic computer graphics images, and one set of 800 images obtained by recapturing the photorealistic computer graphics images using a camera. This dataset is also open for research purposes. Besides the dataset, we also deployed an online demo system [68] for classifying photographic images and computer graphics images, demonstrating our geometric method, as well as the wavelet method [55] and the cartoon method [39] in the prior work. The online demo system is accessible from [www.ee.columbia.edu/trustfoto](http://www.ee.columbia.edu/trustfoto).

In Chapter 4, we present a geometric method for estimating camera response function (CRF) from a single image. The nonlinear CRF is one of the design criteria for a model of camera, in order to mimic the nonlinearities in film and the response of the human visual system, so that the output images are visually pleasing [34]. Therefore, CRF can be considered as a signature for a model of camera. Our geometric method can be applied for distinguishing different models of camera, which belongs to an intermediate-level image source identification problem. We propose a set of geometry invariants which are functions of CRF and invariant to

the locally planar geometry of image irradiance. Through the geometry invariants, CRF can be estimated, regardless of the locally planar geometry. Apart from CRF estimation on a single image, our method exploits the availability of multiple images from the same model of camera to provide a better estimation accuracy and stability. In addition, we propose a generalized gamma curve CRF model (GGCM), which empirically fits the real-world CRF's from DoRF dataset [34] well.

In Chapter 5, we describe other related works in PBIF, which are more loosely related to the works described in this dissertation. We divide the work in PBIF into four major areas according to Fig. 1.4:

1. Image forgery detection (*Is this image authentic?*).
2. Image source identification (*From what device the image is produced?*)
3. Image operation detection (*What post-processing has this image undergone?*)
4. Counter-attack measure design (*How to handle a security attack?*)

Finally, we present the conclusions and the future work of this dissertation in Chapter 6.

This dissertation contains Appendix A through I. Appendix A and Appendix B provides the proof for the two propositions regarding the effect of image splicing on the magnitude and phase responses of bicoherence in Chapter 2. Appendix C provides the mathematical derivation for the gradient operator on a manifold. Appendix D describes the implementation details of our online demo system for classifying photographic images and computer graphic images. Appendix E through I provides the proof for the propositions and the CRF properties described in Chapter 4.

## Chapter 2

# A Statistical Method for Image Splicing Detection

### 2.1 Introduction

We define image splicing as a cut and paste of an image fragment onto another image without further post-processing. Image splicing is the most basic step in the process of creating a 2D composite image, which is shown in Fig. 2.1 (a) with an illustrative example shown in Fig. 2.1 (b). Although post-processing is often applied to blend the spliced object better to the image background and remove the artifacts from the rough region selection, a spliced image without post-processing can be very realistic for non-experts as shown in Fig. 2.2. Although we only consider image splicing in this work, there are specific works in passive-blind image forensics for detecting certain post-processing operations of an image as a tell-tale sign of image

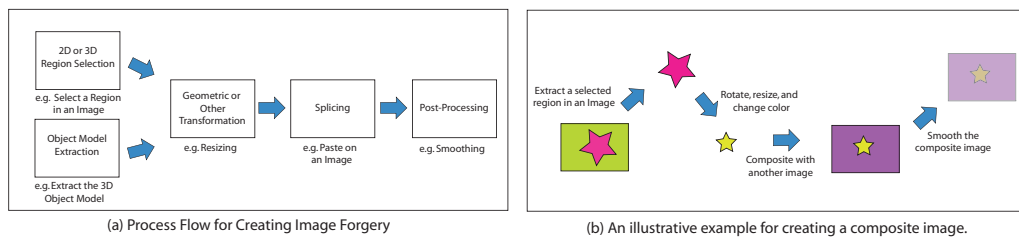


Figure 2.1: (a) The typical image forgery creation process. (b) An illustrative example for creating a composite image.



Figure 2.2: Two examples of spliced images without post-processing where each takes only 10-15 minutes to produce.

compositing [4, 79].

This work is one of the earliest on detecting composite images. The main results of our work are published in [67, 71]. In this work, we propose a statistical method for detecting image splicing by capturing the camera optical low-pass effect on an image. The optical low-pass property is part of the imaging-process authenticity quality of an image. We hypothesize that image splicing introduces sharp edges to the image, which produce a unique magnitude response and phase entropy in the bicoherence measurement computed from the image. We propose a theory to explain the bicoherence response to the abrupt edges and hence, in this image splicing case, offer an alternative to the quadratic phase coupling theory, which is commonly associated with the application of bicoherence.

As natural images originally contain a significant and variable amount of bicoherence energy due to image edges [42], bicoherence-based features are sensitive to the image content, and thus bicoherence features alone are not a robust indicator of the image splicing operation. To overcome this problem, we propose a content-related image abstraction framework that removes such content sensitivity. We abstract an image as its structure component using an image structure-texture decomposition technique [98]. The abstracted image retains the image structure but is devoid of

the splicing artifact. With the new features, the classification accuracy improves by 9%.

## 2.2 Related Works

### 2.2.1 Audio Forgery Detection

In [18], Farid addressed the problem of detecting spliced human speech. They showed that authentic human speech signal is originally weak in the higher-order statistical correlation of its Fourier harmonics and splicing can introduce such a statistical correlation in the signal. They explained the effect of splicing by considering splicing as a non-linear operation, where the quadratic component of the operation induces a higher-order statistical correlation of the Fourier harmonics in the signal. This statistical correlation is called quadratic phase coupling. This statistical correlation can be measured from the magnitude and the phase responses of the signal's bicoherence [41]. In the experiment, they tested the bicoherence technique on 20 authentic human speech sequences, one computer generated speech sequence, and one sequence obtained by splicing human speech sequences in the multi-resolution Laplacian pyramid domain. They showed that the spliced sequence can be separated from others.

### 2.2.2 Image Splicing Detection

There are works [10, 30] that consider the artifacts from image splicing and those from steganography (information hiding) to be similar and apply steganalysis techniques for detecting image splicing. In [30], splicing artifacts are considered non-stationary in a signal and Hilbert-Huang transform is used to extract the non-stationary characteristics of a signal. They also include features from moments of



the wavelet characteristic functions. With the combined features of 110 dimensions, their method achieves a classification accuracy of 80.15% on the splicing benchmark dataset we released in [66]. In comparison, the accuracy we achieved in [71] over the same dataset was 71%. In [10], a spliced image is modeled as having sharp transitions at the splicing interface as proposed in our work. They apply phase congruency to detect the sharp edge transitions. Similarly, they include features from moments of the wavelet characteristic functions. With the combined features of 120D, their method achieves an improved classification accuracy of 82.32% on our splicing dataset.

### 2.3 Bicoherence

In this section, we define the statistical quantity, bicoherence, which is the base feature used in our splicing detection method presented in this chapter. We will denote the spatial variable and the frequency variable respectively as  $x$  and  $\omega$ . A 1D spatial domain signal is represented by a lower-case letter, such as  $a(x)$ , and its Fourier transform is represented by the corresponding upper-case letter, such as  $A(\omega)$ . We reserve the letter  $a$  for an authentic signal,  $d$  as a bipolar signal, and  $s$  for a spliced signal. The bicoherence of a signal  $a(x)$  with a Fourier transform  $A(\omega)$  is denoted by  $B_A(\omega_1, \omega_2)$ . For complex algebra, we denote the magnitude of a complex variable  $z$  as  $|z|$ , its phase as  $\phi(z)$ , and its complex conjugate as  $z^*$ .

**Definition 1** (Bicoherence). *Bicoherence [41] of a 1D signal  $f(x)$  is defined as a third-order moment spectral normalized by its Cauchy-Schwartz upper bound as below:*

$$B_F(\omega_1, \omega_2) = \frac{E [F(\omega_1)F(\omega_2)F^*(\omega_1 + \omega_2)]}{\sqrt{E [ |F(\omega_1)F(\omega_2)|^2 ] E [ |F(\omega_1 + \omega_2)|^2 ]}} \quad (2.1)$$

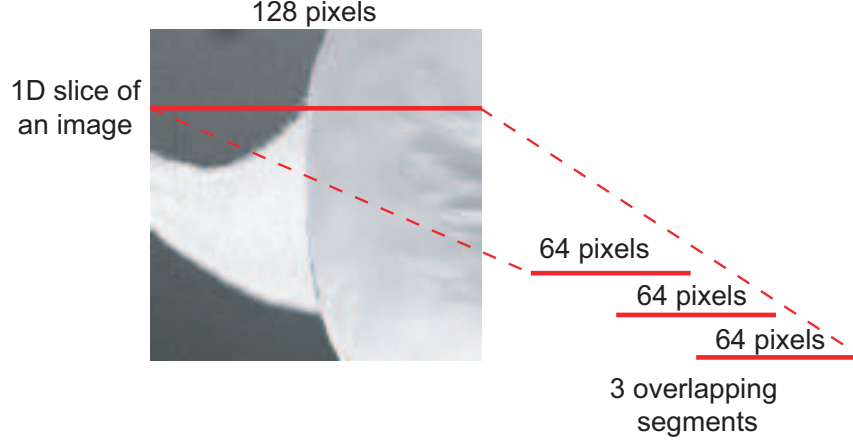


Figure 2.3: 1D slices are extracted from a 2D image. A 1D slice is broken into overlapping segments for estimating bicoherence with Equ. 2.2.

where  $|B_F(\omega_1, \omega_2)| \in [0, 1]$  and  $E(\cdot)$  is the expected value of a random variable.

In practice, for a 1D signal  $f(x)$ , bicoherence is often estimated by sample averaging over the overlapping finite-length segments sampled from the signal using Equ. 2.2:

$$B_F(\omega_1, \omega_2) = \frac{\frac{1}{N} \sum_i (F_i(\omega_1) F_i(\omega_2) F_i^*(\omega_1 + \omega_2))}{\sqrt{\frac{1}{N} \sum_i (|F_i(\omega_1) F_i(\omega_2)|^2) \frac{1}{N} \sum_i (|F_i(\omega_1 + \omega_2)|^2)}} \quad (2.2)$$

In Equ. 2.2, a signal is decomposed into  $N$  overlapping segments, where a segment is denoted by  $F_i(\omega)$ , with  $i = 1, \dots, N$ . Fig. 2.3 shows how overlapping segments are obtained from a 1D slice of an image.

Unlike the power spectrum (i.e., a second-order moment spectral), bicoherence is a complex function and captures the phase information of a signal. Bicoherence has a unique property that its magnitude will achieve the maximal value 1 and its phase will become 0 at a quadratically phase coupled frequency pair  $(\omega_1, \omega_2)$ . *Quadratic phase coupling* (QPC) happens when there exists significant harmonics at the frequencies  $\omega_1$ ,  $\omega_2$ , and  $\omega_1 + \omega_2$ , with their respective phase being  $\phi_1$ ,  $\phi_2$ , and

$\phi_1 + \phi_2$ . QPC phenomena can be found in many measured signals in nature, such as electroencephalogram (EEG) signals [87], human speech signals [17], ocean wave interaction signals [41] and so on. QPC can be induced when a signal undergoes a quadratic-linear operation, which is the lower-order component of a non-linear process. We illustrate the effect of a quadratic-linear operation using a simple 1D signal with two frequency harmonics [18, 73]:

$$r(x) = a_1 \cos(\omega_1 x + \phi_1) + a_2 \cos(\omega_2 x + \phi_2) \quad (2.3)$$

When a linear-quadratic operation with a constant  $\alpha$  is applied on  $r(x)$ , we obtain:

$$\begin{aligned} r(x) + \alpha r(x)^2 &= a_1 \cos(\omega_1 x + \phi_1) + a_2 \cos(\omega_2 x + \phi_2) \\ &+ a_1 a_2 \cos((\omega_1 + \omega_2)x + (\phi_1 + \phi_2)) \\ &+ \frac{1}{2} \alpha a_1^2 \cos(2\omega_1 x + 2\phi_1) + \frac{1}{2} \alpha a_2^2 \cos(2\omega_2 x + 2\phi_2) \\ &+ a_1 a_2 \cos((\omega_1 - \omega_2)x + (\phi_1 - \phi_2)) + \frac{1}{2} \alpha a_1^2 + \frac{1}{2} \alpha a_2^2 \end{aligned} \quad (2.4)$$

Note that the phase of the harmonics at  $\omega_1$ ,  $\omega_2$  and  $\omega_1 + \omega_2$  are respectively  $\phi_1$ ,  $\phi_2$  and  $\phi_1 + \phi_2$ .

The work in [18] considers the splicing operation as being non-linear and hence capable of inducing QPC through the quadratic-linear component of the non-linear image splicing. Such an explanation is offered for the bicoherence capability in splicing detection. However, the quadratic-linear operation as illustrated above is a point-wise function, while the splicing operation is not. Therefore, such a justification through QPC is incomplete.

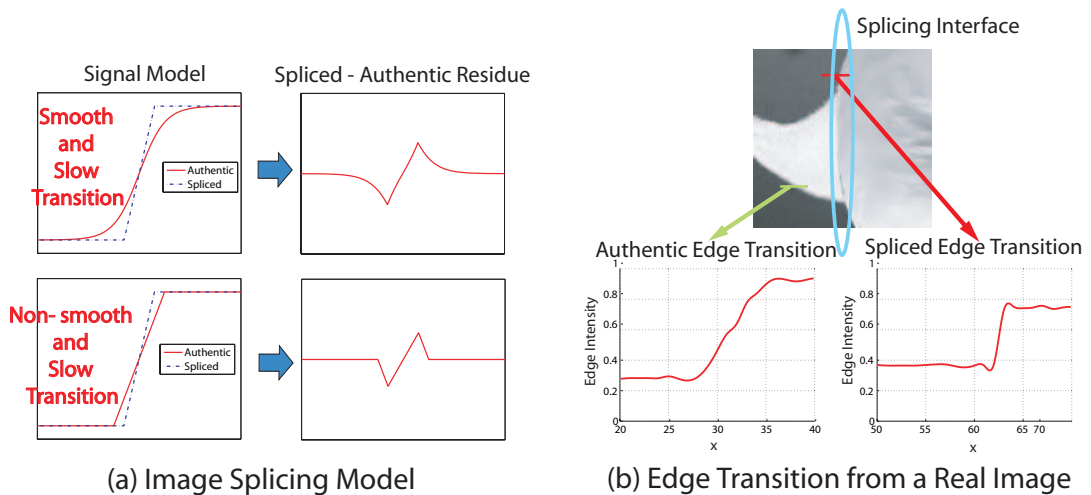


Figure 2.4: (a) The model for an authentic and a spliced signal. (b) The bi-mode residue signal from the difference of an authentic signal and a spliced signal.

## 2.4 Bicoherence Theory for Splicing Detection

We propose an alternative theory to explain the capability of bicoherence in detecting splicing, in terms of its magnitude response and its phase entropy. We consider the splicing operation as an addition of a bipolar signal to the source signal, and hence make a connection to the camera optical low-pass property. The theory predicts that image splicing induces a concentration of the bicoherence phase at  $\pm 90^\circ$ , instead of the  $0^\circ$  from the QPC theory (as described above). We validate this prediction on a real-world splicing benchmark dataset.

### 2.4.1 Splicing Model

When an image fragment is pasted onto another image without post-processing, it is likely to produce both non-smooth and sharp transitions of image intensity at the splicing interface. Such an edge transition is not natural to an authentic image as cameras often have an optical low-pass property for the purpose of anti-aliasing. Fig 2.4 (a) shows the abstract 1D model for the authentic and the spliced

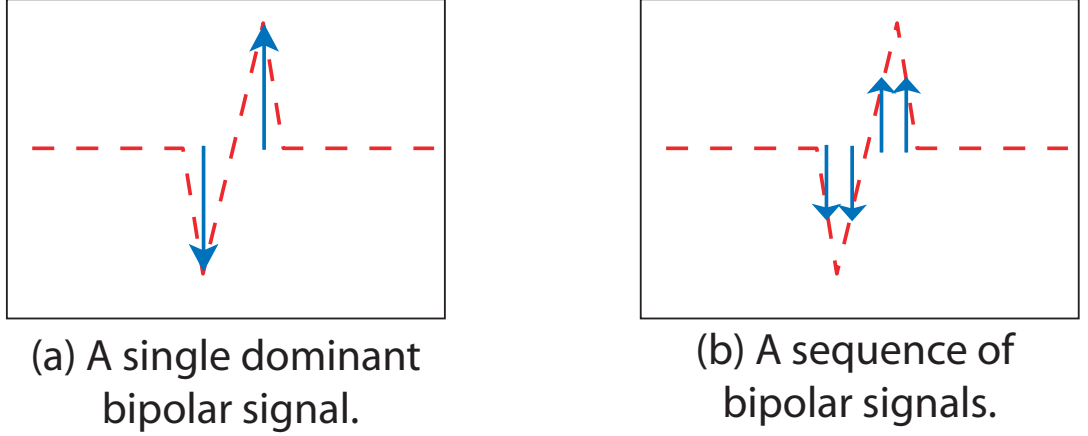


Figure 2.5: A bi-mode residue signal can be approximated as (a) a single dominant bipolar signal. (b) a sequence of bipolar signals.

signal. A bi-mode residue signal is obtained when taking the difference of the two signals. Note that the bi-mode residue is observed even when the authentic signal is non-smooth. Therefore, the sharp edge transition is responsible for the bi-mode residue. Fig. 2.4 (b) shows the authentic and the spliced edge transition in a real image example. Note that the  $x$ -axis of the two plots are at the same scale and by comparison the authentic edge transition is much slower than the spliced one. Due to this observation, we can model a spliced signal as an authentic signal with an additive bi-mode signal, which can be approximated by a single dominant bipolar signal (see Fig. 2.5 (a)) or a sequence of bipolar signals (see Fig. 2.5 (b)).

A bipolar signal can be defined as below:

**Definition 2** (Bipolar Signal Model). *A bipolar signal  $d(x)$  consists of two Dirac delta functions with different polarity and has a Fourier transform  $D(\omega)$ :*

$$d(x) = k (\delta(x - x_o) - \delta(x - x_o - \Delta)) \Leftrightarrow D(\omega) = k (e^{-jx_o\omega} - e^{-j(x_o+\Delta)\omega}) \quad (2.5)$$

where  $\delta(\cdot)$  represents a Dirac delta function,  $\Delta$  is the separation between the two

*Dirac delta functions,  $x_o$  represents the location of the bipolar signal, and  $k \in \mathbb{R}$  is the magnitude of a bipolar signal. The polarity of a bipolar is determined by the sign of  $k$ .*

To simplify our analysis hereforth, we only model splicing with a single additive bipolar signal:

**Definition 3** (Spliced Signal Model). *A spliced signal  $s(x)$  is modeled as*

$$s(x) = a(x) + d(x) \Leftrightarrow S(\omega) = A(\omega) + D(\omega) \quad (2.6)$$

*where  $a(x)$  is an authentic signal.*

In practice, we compute bicoherence through sample averaging of the overlapping finite-length segments sampled from the original 1D signal as in Equ. 2.2. In our analysis, we assume that there can only be at most one bipolar signal for an overlapping segment and the bipolar signals found in the overlapping segments of a signal are the same; that is having the same polarity  $k$  and delta separation  $\Delta$ . Note that, although the location of the bipolar signal,  $x_o$ , may not be the same for different sequences,  $x_o$  has no significance in the computation of Equ. 2.2, as the term with  $x_o$  will be canceled out during the computation. As we assume that there can only be at most one bipolar signal for an overlapping segment, we can model the appearance of a bipolar signal in an overlapping segment using a Bernoulli probability model with the probability of seeing a bipolar signal in an overlapped signal being  $p_d$ .

Note that our above assumption is true for the simplest case of a single splicing interface. By fixing  $k$  and  $\Delta$ , a bipolar signal becomes deterministic, as far the computation of Equ. 2.2 is concerned. Below, we analyze the phase and the magnitude

of the bicoherence for a bipolar signal.

**Property 1** (Phase of Bipolar Signal Bicoherence). *Let  $B_D(\omega_1, \omega_2)$  be the bicoherence of a bipolar signal, then the phase of the bipolar signal bicoherence,  $\phi(B_D(\omega_1, \omega_2)) = \pm 90^\circ$ , where the plus and minus sign of  $\phi(B_D(\omega_1, \omega_2))$  depends on the sign of  $B_D(\omega_1, \omega_2)$ .*

*Proof.*  $D(\omega)$  can be written in two forms:

$$D(\omega) = ke^{-jx_o\omega}(1 - e^{-j\Delta\omega}) \quad (2.7)$$

and

$$D(\omega) = ke^{-jx_o\omega}e^{-j\frac{1}{2}\Delta\omega}(e^{j\frac{1}{2}\Delta\omega} - e^{-j\frac{1}{2}\Delta\omega}) = ke^{-jx_o\omega}e^{-j\frac{1}{2}\Delta\omega} \sin\left(\frac{1}{2}\Delta\omega\right) \quad (2.8)$$

The phase of the bipolar signal bicoherence  $B_D(\omega_1, \omega_2)$  is determined by the bicoherence numerator, which can be expressed in the form of both Equ. 2.9 (derived from Equ. 2.7) and Equ. 2.10 (derived from Equ. 2.8):

$$\begin{aligned} E[D(\omega_1)D(\omega_2)D^*(\omega_1 + \omega_2)] \\ = 2jp_d k^3 (\sin(\Delta\omega_1) + \sin(\Delta\omega_2) - \sin(\Delta(\omega_1 + \omega_2))) \end{aligned} \quad (2.9)$$

$$= 8jp_d k^3 \sin\left(\frac{1}{2}\Delta\omega_1\right) \sin\left(\frac{1}{2}\Delta\omega_2\right) \sin\left(\frac{1}{2}\Delta(\omega_1 + \omega_2)\right) \quad (2.10)$$

where  $p_d$  is the probability of seeing a bipolar signal in an overlapped signal segment

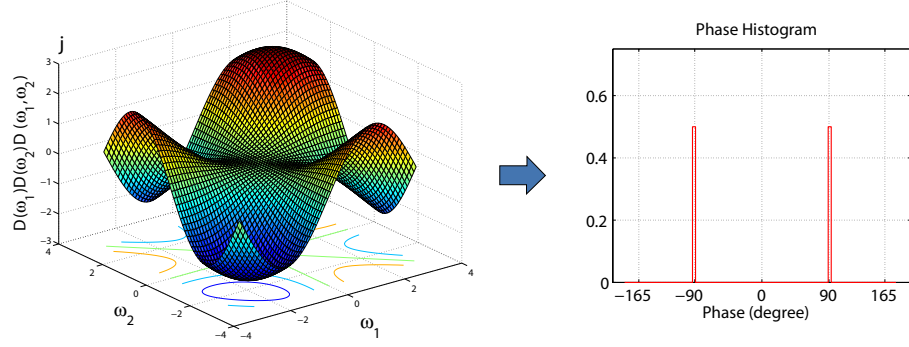


Figure 2.6: A 3D plot for the numerator of the bipolar signal bicoherence for the case of  $k = 1$  and  $\Delta = 1$ . Note that the vertical axis is imaginary and therefore the resulting phase histogram has values concentrated at  $\pm 90^\circ$

used for estimating bicoherence. Then, the phase of a bipolar signal:

$$\begin{aligned}
 \phi(B_D(\omega_1, \omega_2)) &= \phi(E[D(\omega_1)D(\omega_2)D^*(\omega_1 + \omega_2)]) \\
 &= \text{sgn}(8jp_d k^3 \sin(\frac{1}{2}\Delta\omega_1) \sin(\frac{1}{2}\Delta\omega_1) \sin(\frac{1}{2}\Delta(\omega_1 + \omega_2)))\phi(j) \\
 &= \pm 90^\circ
 \end{aligned} \tag{2.11}$$

where the plus and minus sign of  $\phi(B_D(\omega_1, \omega_2))$  is determined by

$$\text{sgn}(8jp_d k^3 \sin(\frac{1}{2}\Delta\omega_1) \sin(\frac{1}{2}\Delta\omega_1) \sin(\frac{1}{2}\Delta(\omega_1 + \omega_2)))$$

with  $\text{sgn}(\cdot)$  being the sign operator. □

Fig. 2.6 shows an example of  $D(\omega_1)D(\omega_2)D^*(\omega_1 + \omega_2)$  and its resulting phase histogram concentrates at  $\pm 90^\circ$ . Note that the phase histogram is symmetric due to the following property:

**Property 2** (Symmetry of Bioherence Phase Histogram). *For a real-valued signal  $f(x)$ , the phase histogram of  $B_F(\omega_1, \omega_2)$  is symmetric.*

*Proof.* For a real-valued signal  $f(x)$ , its Fourier transform is conjugate symmetric,



i.e.,  $F(\omega) = F^*(-\omega)$ , where  $|F(\omega)| = |F(-\omega)|$  and  $\phi(F(\omega)) = -\phi(F(-\omega))$ . Then,

$$B_F^*(-\omega_1, -\omega_2) = \frac{E^* [F(-\omega_1)F(-\omega_2)F^*(-\omega_1 - \omega_2)]}{\sqrt{E [|F(-\omega_1)F(-\omega_2)|^2] E [|F(-\omega_1 - \omega_2)|^2]}} \quad (2.12)$$

$$= \frac{E^* [F^*(\omega_1)F^*(\omega_2)F(\omega_1 + \omega_2)]}{\sqrt{E [|F^*(\omega_1)F^*(\omega_2)|^2] E [|F^*(\omega_1 + \omega_2)|^2]}} \quad (2.13)$$

$$= \frac{E [F(\omega_1)F(\omega_2)F^*(\omega_1 + \omega_2)]}{\sqrt{E [|F(\omega_1)F(\omega_2)|^2] E [|F(\omega_1 + \omega_2)|^2]}} = B_F(\omega_1, \omega_2) \quad (2.14)$$

which implies that  $B_F(\omega_1, \omega_2)$  is also conjugate symmetric. As for every  $\phi(B_F(\omega_1, \omega_2))$ , there is  $\phi(B_F(-\omega_1, -\omega_2)) = -\phi(B_F(\omega_1, \omega_2))$ , therefore the bicoherence phase histogram is symmetric.  $\square$

Below we provide the magnitude property of the bipolar bicoherence:

**Property 3** (Magnitude of Bipolar Signal Bicoherence). *The magnitude of the bipolar signal bicoherence achieves the maximal value of 1 for every  $(\omega_1, \omega_2)$ . As a result, the mean of the bicoherence magnitude over all  $(\omega_1, \omega_2)$  also achieves the maximal value of 1.*

*Proof.* We can write  $D(\omega_1)D(\omega_2) = c(\omega_1, \omega_2)D(\omega_1 + \omega_2)$  where

$$c(\omega_1, \omega_2) = \frac{k(1 - e^{-j\Delta\omega_1} - e^{-j\Delta\omega_2} + e^{-j\Delta(\omega_1 + \omega_2)})}{(1 - e^{-j\Delta(\omega_1 + \omega_2)})} \quad (2.15)$$

Note that  $c(\omega_1, \omega_2)$  is a deterministic when  $k$  and  $\Delta$  are deterministic. If  $p_d$  is the probability of seeing a bipolar signal in the overlapping signal segments, then we

have:

$$|B_D(\omega_1, \omega_2)| = \frac{|E [D(\omega_1)D(\omega_2)D^*(\omega_1 + \omega_2)]|}{\sqrt{E [|D(\omega_1)D(\omega_2)|^2] E [|D(\omega_1 + \omega_2)|^2]}} \quad (2.16)$$

$$= \frac{|p_d D(\omega_1)D(\omega_2)D^*(\omega_1 + \omega_2)|}{\sqrt{p_d |D(\omega_1)D(\omega_2)|^2 p_d |D(\omega_1 + \omega_2)|^2}} \quad (2.17)$$

$$= \frac{|c(\omega_1, \omega_2)|}{\sqrt{|c(\omega_1, \omega_2)|^2}} = 1 \quad (2.18)$$

□

Property 1 and Property 3 describes the effect of a bipolar signal on the magnitude response and the phase entropy for bicoherence. In the following analysis, we will show that the same effect propagates to a spliced signal through a bipolar signal perturbation. Note that, in contrast to the QPC theory which predicts a  $0^\circ$  phase concentration, Proposition 1 predicts a phase concentration of  $\pm 90^\circ$  for a spliced signal.

**Proposition 1** (Bipolar Effect on the Phase of the Spliced Signal Bicoherence). *A bipolar signal induces a  $\pm 90^\circ$  phase concentration on a spliced signal. The strength of the effect depends on the magnitude of  $k$  with respect to the authentic signal energy, and  $p_d$ , the probability of seeing a bipolar signal in an overlapped signal segment used for estimating bicoherence.*

The proof for Proposition 1 is given in Appendix A.

**Proposition 2** (Bipolar Effect on the Magnitude of the Spliced Signal Bicoherence). *An additive bipolar signal induces an increase in the magnitude of the spliced signal bicoherence. The strength of the effect depends on the magnitude of  $k$  with respect to the authentic signal energy, and  $p_d$ , the probability of seeing a bipolar signal in an overlapped signal segment used for estimating bicoherence.*

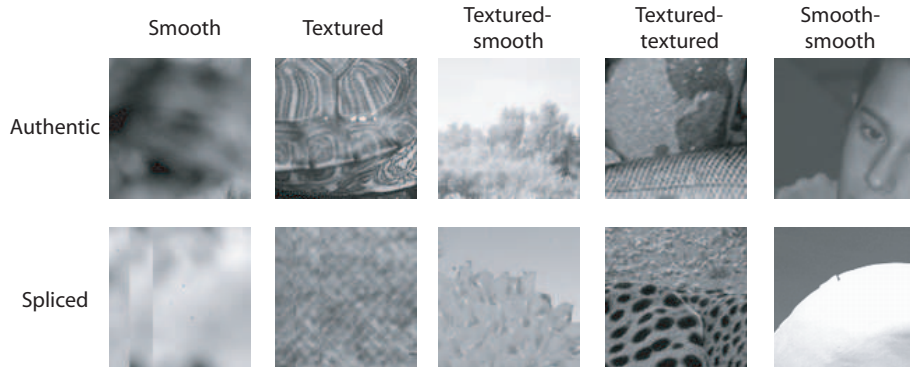


Figure 2.7: Example images from the *Columbia Image Splicing Detection Evaluation Dataset*.

The proof for Proposition 1 is given in Appendix B.

## 2.5 Theory Validation

### 2.5.1 Columbia Image Splicing Detection Evaluation Dataset

To validate the proposed theory on real-world spliced images, we created the *Columbia Image Splicing Detection Evaluation Dataset* [66] which consists of 933 authentic and 912 spliced grayscale image blocks of size  $128 \times 128$  pixels. We focus on splicing detection for such a small-size  $128 \times 128$ -pixel image block instead of the entire image so that localization of splicing interfaces is possible within a given large-size image. The image blocks are created from images in the CalPhotos [8] image set and also using a few images we acquired using a digital camera. The original images are of sizes ranging from  $1800 \times 1000$  pixels to  $800 \times 600$  pixels and are stored in JPEG format. There are five categories of image blocks in the splicing dataset (see Fig. 2.7); they are image blocks of a homogenous textured pattern (denoted as *textured*), a homogenous smooth pattern (*smooth*), and three other categories with an object boundary respectively separating a smooth region from another smooth region (*smooth-smooth*), a textured region from a smooth region (*textured-smooth*),

and textured region from another textured region (*textured-textured*). For the image blocks with an object boundary, the boundary can be either an authentic object boundary or a splicing interface. The procedure for creating the *smooth-smooth*, the *textured-smooth*, and the *textured-textured* image blocks is as follows using Adobe Photoshop:

1. To create a spliced image, we cut an object from an original images along its boundary and then paste the cut-out onto another original image. We repeat this cut-and-paste process for multiple image pairs.
2. To generate the image blocks, we crop the  $128 \times 128$ -pixel image block from the spliced images such that the spliced image block contains a portion of the splicing boundary, or from the authentic images such that the authentic image block contains a portion of the object boundary.
3. The image blocks are saved in an uncompressed BMP format.

We also created spliced image blocks for the *textured* and *smooth* categories by copying a vertical or horizontal strip of 20-pixel wide from one location to another location within the same image block.

### 2.5.2 Computation of Bicoherence Features

To validate the theory, we compute the phase histogram, the phase entropy and the magnitude response of the bicoherence of the image blocks. Given a  $128 \times 128$ -pixel image block in the dataset, we compute bicoherence individually on the horizontal and the vertical slices, each being a 128-pixel long 1D signal. The same computational procedure has been employed in [19]. For each slice, bicoherence is estimated from three 64-pixel long overlapping segments where the overlap is 32-pixel long (see

Fig. 2.3). When computing the discrete Fourier transform, the segments are multiplied with a Hanning window and extended to a length of 128 pixels by zero-padding, in order to reduce frequency leakage and obtain a better frequency resolution. We also experimented with a shorter 32-pixel long overlapping segments, which gives us very similar experimental results as those presented in this chapter.

The bicoherence phase histogram  $p(\phi_i)$ ,  $i = 1, \dots, 24$  for each slice is obtained by uniformly binning of the bicoherence phase (from  $-180^\circ$  to  $180^\circ$ ) into 24 bins (bin width =  $15^\circ$ ). The phase histogram is then normalized so that it sums to 1. The overall phase histogram for an image block is obtained by averaging the phase histogram of the 1D slices.

The magnitude response  $r_m$  and the phase entropy  $r_p$  of bicoherence is estimated as below:

$$r_m = \frac{1}{|\Omega|} \sum_{(\omega_1, \omega_2) \in \Omega} B(\omega_1, \omega_2), \quad r_p = \sum_i p(\phi_i) \log(p(\phi_i)) \quad (2.19)$$

$r_m$  represents the averaged magnitude over all  $(\omega_1, \omega_2)$  and  $r_p$  is the negative entropy of the bicoherence phase, where a higher  $r_p$  represents a higher concentration of the bicoherence phase (i.e., less uniformly distributed). The overall magnitude response  $R_m$  and the overall phase entropy  $R_p$  of an image block is obtained by:

$$R_m = \sqrt{\left(\frac{1}{128} \sum_{i=1}^{128} r_m^{vi}\right)^2 + \left(\frac{1}{128} \sum_{i=1}^{128} r_m^{hi}\right)^2}, \quad R_p = \sqrt{\left(\frac{1}{128} \sum_{i=1}^{128} r_p^{vi}\right)^2 + \left(\frac{1}{128} \sum_{i=1}^{128} r_p^{hi}\right)^2} \quad (2.20)$$

where  $r_m^{vi}$  and  $r_m^{hi}$  are respectively the bicoherence magnitude response of the vertical and the horizontal slices, indexed by  $i$ , while  $r_p^{vi}$  and  $r_p^{hi}$  are those for the phase.

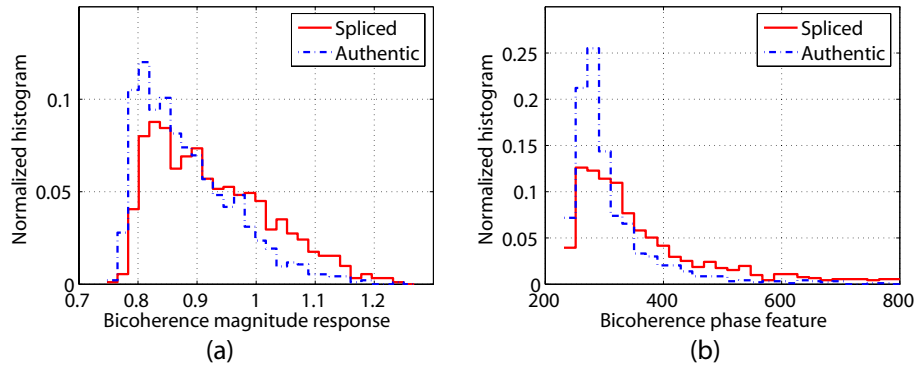


Figure 2.8: The distribution of (a) the magnitude response  $R_m$  and (b) the phase entropy  $R_p$  for bicoherence computed on the Columbia Image Splicing Detection Evaluation Dataset.

### 2.5.3 Experiments

We computed the bicoherence magnitude response  $R_m$  and the bicoherence phase entropy  $R_p$  for the image blocks in our splicing dataset. The distributions of the  $R_m$  and  $R_p$  for the authentic image blocks and the spliced image blocks are shown in Fig. 2.8. For both  $R_m$  and  $R_p$ , the distributions for the authentic and the spliced image block are verified to be different by a Kolmogorov-Smirnov test with a 5% significance level. Note that both  $R_m$  and  $R_p$  for the spliced image blocks are higher than those of the authentic image blocks in terms of distribution. A higher  $R_p$  represents a higher concentration of the bicoherence phase. These empirical results match the prediction of Proposition 1 and Proposition 2.

Fig. 2.9 (a) shows the averaged phase histogram for the spliced and the authentic image blocks. Note that the spliced block phase histogram is higher than the authentic one at  $\pm 90^\circ$  but lower at  $0^\circ$ . This observation is more clearly shown in Fig. 2.9 (b) by the difference of the two histograms, which matches the prediction of Proposition 1.

We perform a baseline classification of the authentic and the spliced image blocks

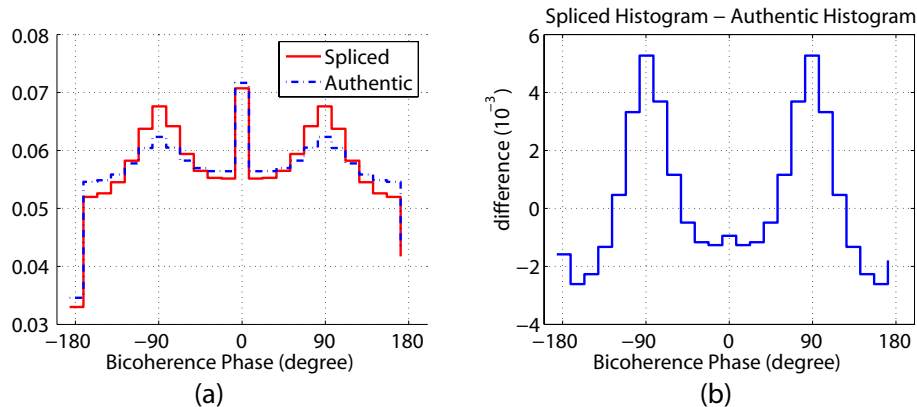


Figure 2.9: (a) The average bicoherence phase histogram for the authentic image blocks and the spliced image blocks in the Columbia Image Splicing Detection Evaluation Dataset. (b) The difference histogram obtained by subtracting the authentic image block phase histogram from that of the spliced image blocks.

with  $R_m$  and  $R_p$  using a Support Vector Machine (SVM) classifier of the LIBSVM [37] implementation. We use the Radial Basis Function (RBF) kernel for the SVM and model selection (for the regularization parameter  $C$  and the kernel parameter  $\gamma$ ) is done by a grid search [37] in the joint parameter space,  $(C, \gamma)$ . The classification accuracy for the binary classes, obtained by a five-fold cross-validation procedure, is 63.6%, which is about 13% better than random guessing. The specific information about  $\pm 90^\circ$  phase concentration does not help much, as the phase concentration response has already been captured by  $R_p$ . The poor classification performance using the bicoherence features can be explained by an observation given in [42]. In [42], it is observed that natural-scene images originally contain a significant amount of energy in higher-order statistics including bicoherence, due to image edges. As the original amount of the image-edge related bicoherence energy is variable, it makes the increase in the bicoherence magnitude due to image splicing less detectable. Although the work in [42] does not provide information about the bicoherence phase of natural-scene images, we conjecture that the edge

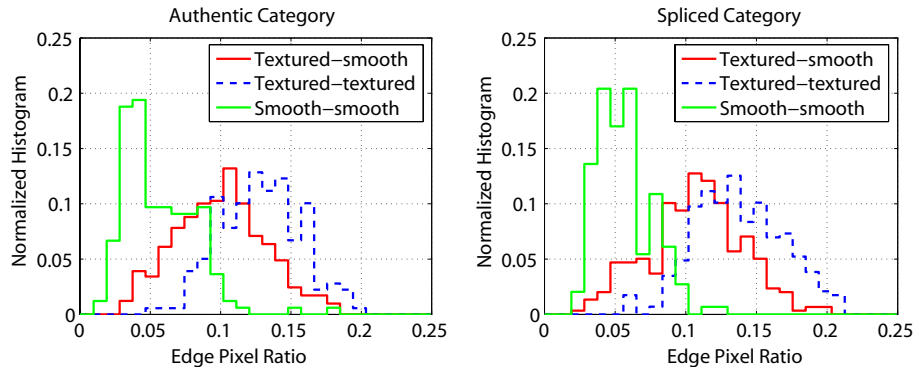


Figure 2.10: The distribution of the edge pixel ratio feature for the authentic and the spliced image blocks in our dataset.

structure in natural-scene images induces  $\pm 90^\circ$  phase concentration to a certain extent. The conjecture is based on the observation that the averaged bicoherence phase histogram for the authentic image blocks shown in Fig. 2.9 (a) exhibits a  $\pm 90^\circ$  phase concentration. Therefore, the bicoherence phase feature encounters the same problems as the bicoherence magnitude feature does. In the next subsection, we propose new image features to improve the classification performance.

## 2.6 Improvement on Splicing Detection

As it is observed in [42] that the original amount of bicoherence energy is related to image edge or more generally image content, so we can improve the classification performance of the bicoherence features by importing image-content-related features. We propose two image-content-related features, which are the edge pixel ratio feature  $E_r$ , and the image structure bicoherence features  $(U_m, U_p)$ .

### 2.6.1 Edge Pixel Ratio Feature

The edge pixel ratio feature  $E_r$  is the proportion of image pixels being edge pixels, for instance those detected by the Canny edge detection algorithm [32]. This feature



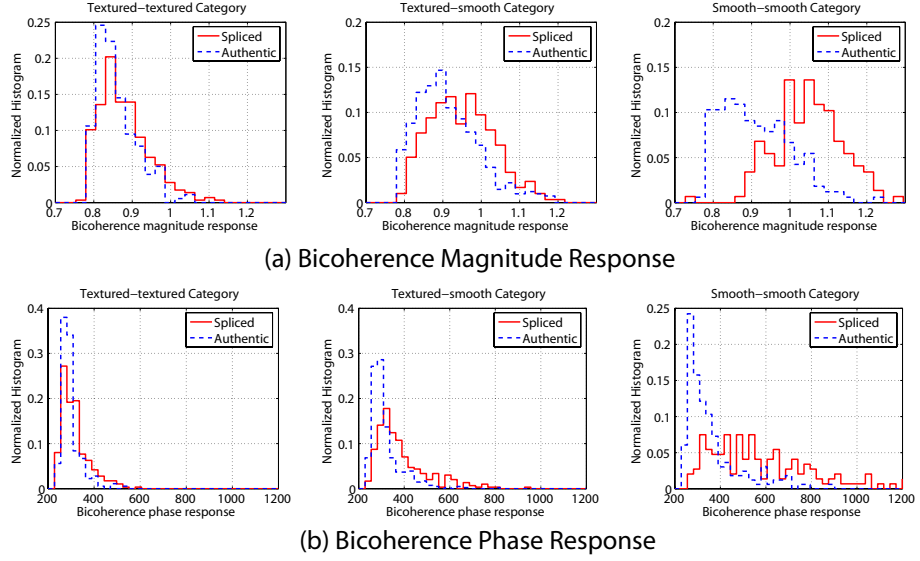


Figure 2.11: The distribution of (a) the bicoherence magnitude response and (b) the bicoherence phase entropy for image blocks of the textured-textured, the textured-smooth, and the smooth-smooth categories in our dataset.

qualitatively measures the textured-ness of an image, as shown in Fig. 2.10, where image blocks of the textured-textured category has the higher edge pixel ratio in terms of the distribution, followed by those in the textured-smooth category and then those in the smooth-smooth category. Note that, the distribution of these categories for both the authentic and the spliced classes are quite similar. In Fig. 2.11, we observe that the distribution of the bicoherence magnitude response and the bicoherence phase entropy for image blocks of the textured-textured, the textured-smooth, and the smooth-smooth categories are quite different. The authentic and the spliced image blocks of the smooth-smooth category are the most distinguishable, followed by those in the textured-smooth category and the textured-textured category. Therefore, with edge pixel ratio as an additional feature, such a difference may be learnt by a classifier.

### 2.6.2 Image Structure Bicoherence Features

The image structure bicoherence features refers to the bicoherence magnitude response  $U_m$  and the bicoherence phase entropy  $U_p$  for the structure component of an image. The structure component (named as a cartoon component in [98]) of an image is obtained by an image structure-texture decomposition algorithm proposed in [98]. In the algorithm, an image  $f(x, y)$  is modeled by a linear model,  $f(x, y) = u(x, y) + v(x, y)$ , where  $u(x, y)$  represents the structure component or a simplification of an image, and  $v(x, y)$  represents the texture component.  $u(x, y)$  is modeled by a function of bounded variation (BV), where sharp edges are permitted, while the texture component of an image is modeled by an oscillatory function. Therefore, the edge structure is better preserved in  $u(x, y)$  compared to a linear low-pass image. The decomposition is achieved in the total variation minimization framework [84]:

$$\inf_u E(u) = \int |\nabla u| dx dy + \lambda \|v\|_* \quad (2.21)$$

where  $\lambda$  is the tuning parameter,  $\|\cdot\|_*$  is a norm inducing a Banach space that allows for oscillating functions. Note that the first term in Equ. 2.21 is a regularizing term, while the second is a fidelity term. The minimizer can be obtained by solving the associated Euler-Lagrange equations, which is a set of partial differential equations (PDE). The solution of the PDE is obtained through a finite difference scheme that is given in [98]. Examples of the structure-texture decomposition are shown in Fig. 2.12, note the extracted texture.

The structure component is only useful for our case if it represents only the image content, but with few or no splicing artifacts, as far as the bicoherence responses are concerned. In other words, we expect most of the splicing artifacts (in terms of sharp edge transitions) will manifest in the texture component. Fig. 2.13 shows the

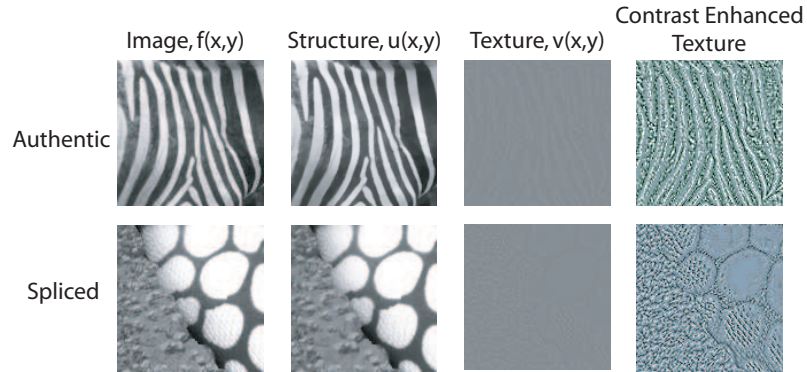


Figure 2.12: Examples for the structure-texture decomposition.

distribution of the bicoherence magnitude response  $U_m$  and the bicoherence phase entropy  $U_p$  for the structure component of the authentic and the spliced image blocks. Although the response distributions for the authentic and the spliced blocks are not exactly the same, they are more similar to each other than the distributions  $R_m$  and  $R_p$  which are illustrated in Fig 2.8. As a result, we can use  $U_m$  and  $U_p$  as a neutral content reference for  $R_m$  and  $R_p$  respectively, in order to make the increase in  $R_m$  and  $R_p$  due to image splicing more prominent.

As an alternative to the structure component, we may compute the bicoherence features from the texture component, which we suspect carries more artifacts from splicing. However, our experiments did not show the effectiveness of this approach. The probable reason is that the effect of the splicing artifacts are less prominent in the noise-like texture component of an image, as far as the bicoherence features are concerned.

## 2.7 Classification Experiment

We evaluate the performance of the proposed edge pixel ratio feature  $E_r$ , and the image structure bicoherence features  $(U_m, U_p)$  by adding them as additional fea-

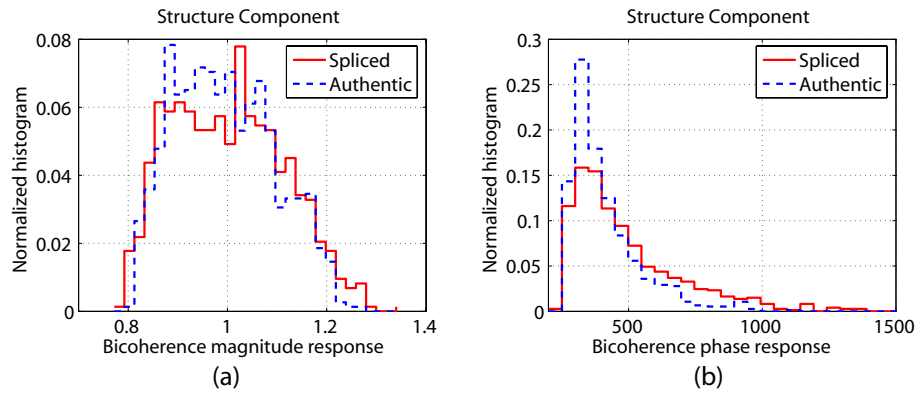


Figure 2.13: The distribution of (a) the bicoherence magnitude response  $U_m$  and (b) the bicoherence phase entropy  $U_p$  for the structure component of the authentic and the spliced image blocks.

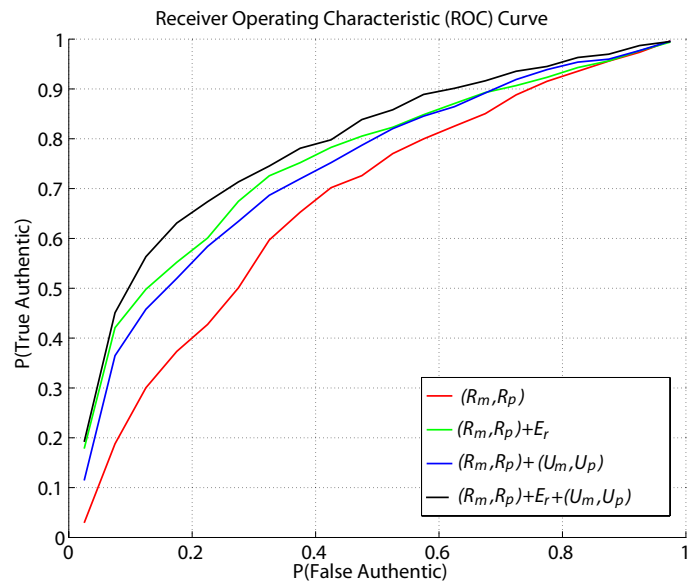


Figure 2.14: The receiver operating characteristic (ROC) curve for the different feature combinations.

Table 2.1: SVM Classification Accuracy for Different Feature Combinations

Features	Accuracy	Features	Accuracy
$(R_m, R_p)$	63.6%	$(R_m, R_p) + E_r$	69.6%
$(R_m, R_p) + (U_m, U_p)$	67.9%	$(R_m, R_p) + E_r + (U_m, U_p)$	72.4%

tures to the basic bicoherence features  $(R_m, R_p)$ , for SVM classification experiments described in Sec. 2.5.3. The averaged binary-class classification accuracy and the classifier ROC curve for different feature combinations are respectively shown in Table 2.1 and Fig. 2.14.

## 2.8 Discussions

Although the classification accuracy of 72.4% with all the features offers an improvement of about 9% on the basic bicoherence features, this accuracy is still considered low for an application concerning image splicing detection. Some recent works [10, 30] have proposed steganalysis-inspired techniques and achieved a better classification accuracy of about 80% (with a different cross-validation setting and data partitioning than our own) on our image splicing dataset, which is encouraging. This dataset is mainly constructed using images from the CalPhotos dataset [8] contributed by many different photographers. These images may not represent the diverse images that we encounter today, therefore, further experiments on a more diverse image dataset is needed. In addition, our experiment only demonstrates the splicing detection accuracy on the  $128 \times 128$ -pixel image blocks, further experiment on the full-size images would offer a more realistic evaluation of image splicing detection. Last but not least, the detection of post-processing operations needs to be considered with image splicing detection under a single framework.

## Chapter 3

# A Geometric Method for Photographic and Computer Graphics Images Classification

### 3.1 Introduction

As model-based computer graphic rendering technology is making great strides, distinguishing photographic image from photorealistic computer graphics is becoming more challenging. A 3D graphic company Autodesk Alias designs an online quiz at [www.fakeorfoto.com](http://www.fakeorfoto.com) for challenging users' visual judgement in distinguishing photographic images (PIM) and photorealistic computer graphics (PRCG). Some of the images from [www.fakeorfoto.com](http://www.fakeorfoto.com) are shown in Fig. 3.1, note the level of photorealism of the PRCG. Furthermore, there are also experimental evidences that, to the naked eye, computer graphic images of certain scenes are visually indistinguishable from photographic images [63].

With high photorealism, PRCG naturally qualifies themselves as potential suspects for forged images. Forged images can be used for fraud, make-belief, and dishonest setup, and the goal of image forensics is to detect these forgery images. In this work, we propose a set of physics-motivated features for distinguishing PIM and PRCG images, for which the main results have been published in [68, 70]. The



Figure 3.1: Examples of photo and PRCG from [www.fakeorfoto.com](http://www.fakeorfoto.com).

physics-motivated features are obtained by studying the respective physical image generative process for PIM and PRCG, and they capture both the scene authenticity and the imaging-process authenticity properties. As a result, we can partially explain the actual differences between PIM and PRCG.

To capture the physical differences in the image generative processes, we propose an image description framework inspired by Mandelbrot. Mandelbrot [57] introduced the idea of fractals as a geometric description of a mathematical object with a fractional dimension to generalize the classical geometry which is limited to integer dimensions. He also pointed out that, unlike the ideal fractal which is a mathematical concept, the geometry of real-world objects are often best characterized by having different dimensions over different range of scales. This insight inspires our image description framework in scale space: at the finest scale, we describe the intensity function of an image as a fractal, while at an intermediate scale as a 2D topological surface with a smooth structure, which is best described in the language of differential geometry. Additionally, we also model the local geometry of the image intensity function in a “non-parametric” manner by local image patches.

Another recent thread of development in image synthesis is image-based rendering [86], where a novel-view image is synthesized from a set of photographic images, or a photorealistic image is obtained when a computer graphic object is given the appearance of a real object which is extracted from photographs of multiple views and lightings. These types of images are commonly found in movies with amazing special visual effects, such as ‘The Lord of the Rings’ (2001), ‘Superman Returns’ (2006) and so on. In essence, images synthesized from image-based rendering are the hybrid of photographic images and computer graphics, hence it is an ambiguous case for our method and beyond the scope of our consideration.

### 3.2 Related Works

Despite the fact that classification of photographic images and computer graphics has been applied for improving the image and video retrieval performance [3, 39, 89], classification of photographic images (PIM) and photorealistic computer graphics (PRCG) is a new problem. The work in [55] takes advantage of the wavelet-based natural image statistics, and extracts the first four order statistics of the in-subband coefficients and those of the cross-subband coefficient prediction errors as features for classifying PIM and PRCG. Promising results, with a PIM detection rate of 67% at a 1% false alarm rate, have been achieved. *However, due to the lack of a physical model for PIM and PRCG, the results have not led to an insight into the question: How is PIM actually different from PRCG?* In [101], an efficient PIM and PRCG classifier is proposed based the characteristic functions of wavelet histograms. With an experiment on a dataset consisting of 4546 online PIM, 3844 online PRCG and the images from our Columbia Photographic Images and Photorealistic Computer Graphics Dataset (see Sec. 3.10), they found that their technique outperforms that



of [55].

### 3.3 Image Generation Process

In general, the image intensity function  $I : (x, y) \in \mathbb{R}^2 \mapsto \mathbb{R}$  arises from a complex interaction of the object geometry, the surface reflectance properties, the illumination and the camera view point. In addition, as photographic or scanned images are captured by an image acquisition device such as a camera or a scanner, they also bear the characteristics of the device (see Fig. 1.5). For example, a digital photographic image in general has undergone the optical lens transformation, the gamma correction, the white-balancing and the color-processing while being tinted with quantization noise and sensor fixed pattern noise [96].

However, PRCG is produced by a graphics rendering pipeline [1], a different process than that of the PIM. In general, a graphics rendering pipeline can be divided into three conceptual stages: application, geometry and rasterizer. At the application stage, mainly implemented in software, the developer designs/composes the objects/scene to be rendered. The objects are represented by the rendering primitives such as points, lines and triangles. The geometry stage, mainly implemented in hardware, consists of rendering operations on the rendering primitives. The rasterizer stage is responsible for converting the rendered primitives into pixels which can be displayed on a screen. During this conversion, the camera effects, such as the depth-of-field (DoF) effect or the gamma correction, may or may not be simulated. The main differences between the PIM and PRCG image generation processes are as follows:

1. **Object Model Difference:** The surface of real-world objects, except for man-made objects, are rarely smooth or of simple geometry. Mandelbrot [57]

has showed the abundance of fractals in nature and also related the formation of fractal surfaces to basic physical processes such as erosion, aggregation and fluid turbulence. However, computer graphics 3D objects are often represented by polygonal models. Although polygonal models can be arbitrarily fine-grained, it comes with a higher cost of memory and computational load. Furthermore, such a polygonal model is not a natural representation for fractal surfaces [76]. A coarse-grained polygonal model may be used at the perceptually insignificant area in an image for saving computational resources.

2. **Surface Model Difference** [1]: The physical light field captured by a camera is a result of the physical light transport from an illumination source, reflected to an image acquisition device by a scene object (see Fig. 1.5). The most general surface reflectance function is 12D, with the parameters being the incoming and the outgoing angle, position, time, and wave length [72]. Precise modeling of a surface reflectance property using the 12D function will make the simulation of light transport computationally expensive, if not infeasible. Therefore, a simplified model based on the assumption of isotropy, spectral independence and parametric representation is often used in computer graphics rendering.
3. **Acquisition Difference**: PIMs carry the characteristics of the imaging process, while PRCG may undergo different types of post-processing after the rasterizer stage. There is no standard set of post-processing techniques, but a few possible ones are the simulation of the camera effect, such as the depth of field, gamma correction, addition of noise, and retouching.

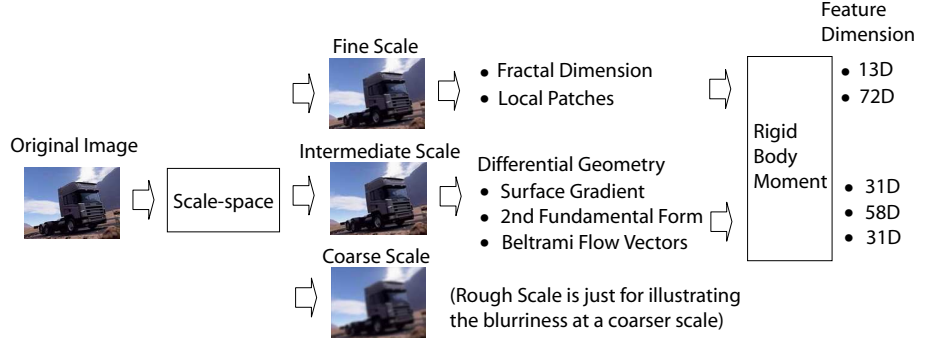


Figure 3.2: The geometry-based image description framework.

### 3.4 Geometry-based Image Description Framework

To exploit the differences between PIM and PRCG, we propose a two-scale image description framework, inspired by Mandelbrot [57] (see Fig. 3.2). At the finest scale, the image intensity function is related to the fine-grained details of a 3D object’s surface properties. The finest-scale geometry can be characterized by the local fractal dimension (Section 3.6) and also by the “non-parametric” local patches (Section 3.8). At an intermediate scale, when the fine-grained details give way to a smoother and differentiable structure, the geometry can be best described in the language of differential geometry, where we compute the surface gradient (Subsection 3.7.1), the second fundamental form (Subsection 3.7.2) and the Beltrami flow vectors (Subsection 3.7.3).

The transition of an image to an intermediate scale is done in the linear Gaussian scale-space (Section 3.5), where a scale-space image is infinitely differentiable.

#### 3.4.1 Notation for This Chapter

In this chapter, a single-channel (grayscale) intensity function is represented as  $I(x, y)$ , while a joint-RGB-color function is represented as  $I^{RGB}(x, y) = (I^R(x, y), I^G(x, y), I^B(x, y))$ . Their corresponding scale-space function are respectively de-

noted as  $L(x, y)$  and  $L^{RGB}(x, y) = (L^R(x, y), L^G(x, y), L^B(x, y))$ . We denote the first-order partial derivative of a function  $f(x, y)$  as  $f_x$  and  $f_y$ , a similar convention for the higher-order partial derivative is applied. When needs arise, a partial derivation operator  $\frac{\partial}{\partial x}$  may be represented as  $\partial_x$  to simplify notation. We will reserve the character  $g$  for the Riemannian metric of a manifold,  $r$  for image irradiance, and  $s$  for the scale parameter in scale space,  $M$  for a manifold or a submanifold.

### 3.5 Linear Gaussian Scale-space

In this section, we will give a brief introduction of the linear Gaussian scale-space in which we compute the local fractal dimension and the differential-geometric quantities. The linear Gaussian scale-space  $L : \Omega \subseteq \mathbb{R}^2 \times \mathbb{R}_+ \mapsto \mathbb{R}$  of a 2D image  $I : \Omega \subseteq \mathbb{R}^2 \mapsto \mathbb{R}$  is given by:

$$L(x, y; s) = \iint_{\Omega} I(\xi, \eta) \phi(x - \xi, y - \eta; s) d\xi d\eta = \phi(x, y; s) * I(x, y) \quad (3.1)$$

where  $L(x, y; 0) = I(x, y)$ ,  $s$  is a non-negative real number called the scale parameter,  $*$  is the convolution operator and  $\phi : \mathbb{R}^2 \mapsto \mathbb{R}$  is the Gaussian kernel function:

$$\phi(x, y; s) = \frac{1}{2\pi s} e^{-\frac{x^2+y^2}{2s}} \quad (3.2)$$

Even though an image,  $I(x, y)$  may not be differentiable initially, the corresponding linear scale-space function,  $L(x, y; s)$ ,  $s > 0$  is infinitely differentiable with respect to  $(x, y)$  as long as  $I(x, y)$  is bounded. As differential-geometric quantities are the composition of derivative terms, the differentiability property ensures that the computed differential-geometric quantities are well-defined. The partial derivative of a scale-space can be obtained by convolving the original image,  $I(x, y)$ , with

the partial derivatives of the Gaussian kernel function  $\phi(x, y; s)$ :

$$L_{x^n y^m}(x, y; s) = \partial_{x^n} \partial_{y^m} (\phi(x, y; s) * I(x, y)) \quad (3.3)$$

$$= (\partial_{x^n} \partial_{y^m} \phi(x, y; s)) * I(x, y) \quad (3.4)$$

### 3.6 Fractal Geometry

The **Object Model Difference** mentioned in Section 3.3 implies that the 3D computer graphic model's surface properties deviate from the real-world object's surface properties, which are associated with the physical formation process such as erosion. This deviation will directly result in a deviation of the local fractal dimension measured from the image intensity function, under certain assumptions [76]. Based on this, we conjecture that the deviation of the surface property would result in a different distribution for the local fractal dimension of PRCG.

In this section, we briefly introduce fractal geometry and the techniques for computing fractal dimension. A fractal is defined as a set of mathematical objects with a fractal dimension (technically known as the Hausdorff Besicovitch dimension) strictly greater than the topological dimension of the object but not greater than the dimension of the Eculidean space where the object lives. For example, a fractal coastline lives on a 2D surface, and, being a line, has a topological dimension of one, then its fractal dimension would be  $1 < D \leq 2$ .

For a real world object, to directly estimate the fractal dimension from the mathematical definition of the Hausdorff Besicovitch dimension is difficult. A fractal is self-similar across scales, so fractal dimension is often estimated as a factor of self-similarity. A commonly used random fractal model for images is called fractional Brownian motion (fBm) [57, 76]. With the fBm model, one method for estimating

the fractal dimension is by measuring the self-similarity factor of a quadratic differential invariant in scale-space. We select this estimation technique in order to keep our approach consistent in the sense that both the fractal geometry and the differential-geometric quantities are computed in the linear scale-space. We herein describe the estimation procedure. We first compute the  $L1$ -norm of the second-order quadratic differential invariant:

$$\|I^{(2)}(s)\| = \sum_{\text{all } (x,y)} |I^{(2)}(x,y;s)| \quad \text{where } I^{(2)} = L_{xx}^2 + 2L_{xy}^2 + L_{yy}^2 \quad (3.5)$$

at multiple scales from  $s = 2^2$  pixels to  $s = 2^8$  pixels with an exponential increment. Then, we perform a least square linear regression on the  $\log \|I^{(2)}(s)\| - \log(s)$  plot and measure the slope of the line. With the estimated slope, the fractal dimension is obtained by  $D = \frac{1}{2} - \text{slope}$ . Fig. 3.3 shows two examples of fractal dimension estimation using the  $\log \|I^{(2)}(s)\| - \log(s)$  plot. Note that a higher fractal dimension for the tree image block indicates a perceptually rougher image function and a more rapid decrease of the quadratic differential invariant. For feature extraction, we compute the fractal dimension for each of the non-overlapping  $64 \times 64$ -pixel local blocks, independently for the R, G and B color channels of an image. As a result, each local block produces a 3D fractal dimension vector across the color channels. For each image, we obtain a distribution of data points in the 3D space.

### 3.7 Differential Geometry

This section introduces three differential-geometric quantities: the surface gradient, the second fundamental form and the Beltrami flow vector, computed in scale space with scale  $s = 1$  pixel. The scale  $s = 1$  pixel is empirically found to produce good

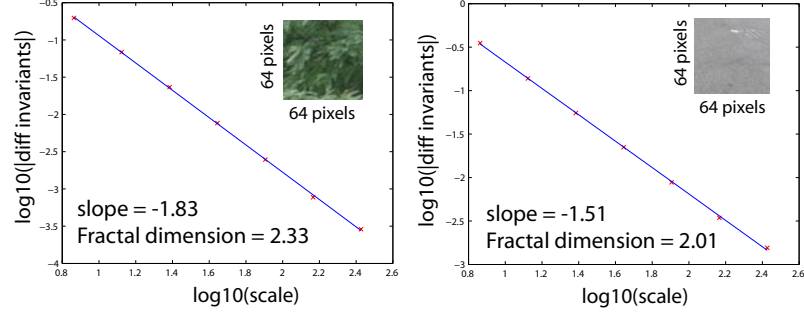


Figure 3.3: Log-log plot for estimating the fractal dimension of a  $64 \times 64$ -pixel block from the tree (Left) and road (Right) region

classification performance.

Having a fixed scale, we will represent a scale-space function as  $L(x, y)$ , without explicitly specifying the scale parameter  $s$ . As functions in scale-space are infinitely differentiable, so are the single-channel intensity function  $L(x, y)$  and the joint-RGB color intensity function  $L^{RGB}(x, y) = (L^R(x, y), L^G(x, y), L^B(x, y))$ . As a result, the graph of  $L(x, y)$  and  $L^{RGB}$  are smooth and respectively defined as:

$$F^I : (x, y) \in \mathbb{R}^2 \mapsto (x, y, L(x, y)) \in \mathbb{R}^3 \quad (3.6)$$

$$F^{RGB} : (x, y) \in \mathbb{R}^2 \mapsto (x, y, L^R(x, y), L^G(x, y), L^B(x, y)) \in \mathbb{R}^5 \quad (3.7)$$

The graph of a smooth function qualifies as a submanifold [45] in Euclidean space, which naturally induces a Riemannian metric on the submanifold. A Riemannian metric  $g$  on a submanifold  $M$  is a symmetric and positive-definite 2-tensor field defined on the tangent plane  $T_p M$  at each point  $p$  on  $M$  [45]. Just like an inner product (also a 2-tensor) in a Euclidean space which allows us to define lengths of vectors and angles between them, a Riemannian metric allows us to define length and angles on the tangent plane of a manifold. With  $g$  defined, the geometry of

the manifold can be measured implicitly, without referring to the ambient space. Therefore,  $g$  is an important element for describing the geometry of a manifold. For a general graph function  $F(x, y)$  (e.g.,  $F : \mathbb{R}^2 \mapsto \mathbb{R}^3$ ),  $g$  can be described by a symmetric and positive-definite matrix:

$$g = \begin{pmatrix} \langle F_x, F_x \rangle & \langle F_x, F_y \rangle \\ \langle F_y, F_x \rangle & \langle F_y, F_y \rangle \end{pmatrix} \quad (3.8)$$

where  $\langle \cdot, \cdot \rangle$  represents the standard inner product in Euclidean space. Hence, for  $F^I(x, y)$  and  $F^{RGB}(x, y)$ , their Riemannian metrics  $g^I$  and  $g^{RGB}$  are respectively given by:

$$g^I = \begin{pmatrix} 1 + L_x^2 & L_x L_y \\ L_x L_y & 1 + L_y^2 \end{pmatrix} \quad (3.9)$$

$$g^{RGB} = \begin{pmatrix} 1 + (L_x^R)^2 + (L_x^G)^2 + (L_x^B)^2 & L_x^R L_y^R + L_x^G L_y^G + L_x^B L_y^B \\ L_x^R L_y^R + L_x^G L_y^G + L_x^B L_y^B & 1 + (L_y^R)^2 + (L_y^G)^2 + (L_y^B)^2 \end{pmatrix} \quad (3.10)$$

The  $g$  matrix can be denoted as  $(g_{ij})$ , where  $g_{ij}$  is an element of  $g$ . Then, with  $g$ , the ‘inner product’ or the first fundamental form on the tangent vectors at a point  $p$  can be written as:

$$\langle V, W \rangle_g = \sum_{ij} g_{ij} V^i W^j \quad (3.11)$$

where  $V = \sum_i V^i e_i$  and  $W = \sum_i W^i e_i$  are tangent vectors at a point, with  $\{e_i\}$  being the tangent plane basis.



### 3.7.1 Gradient on Surface

The **Acquisition Difference**, as mentioned in Section 3.3, can detect PRCG that have not undergone gamma correction as PIM generally do. One reason for missing gamma correction is that popular rendering platforms such as Silicon Graphics use hardware for gamma correction to enhance the contrast of the displayed images, therefore gamma correction on the images is not necessary. Additionally, gamma correction may be performed using the post-processing software such as Adobe Photoshop where the transformation is mainly subjected to the user’s taste. In this section, we will show that the surface gradient of the image intensity function can be used to distinguish PIM and PRCG.

The image intensity function captured by cameras, unless specifically set, has mostly been transformed by a camera response function (CRF), for the purpose of displaying gamma correction as well as for dynamic range compression. A CRF transforms image irradiance from the real-world scene to image intensity  $I(x, y)$ , which is the output of a camera. The typical concave shape of CRF, as shown in Fig. 3.4, is given by the averaged curve from the DoRF database [34] with 201 real-world CRF’s.

One main characteristic of the CRF in Fig. 3.4 is that image irradiance of low values would be stretched and those of high values would be compressed during the non-linear transformation. This effect can be illustrated visually in Fig. 3.5. Let the image intensity function be  $I(x, y) = f(r(x, y)) \simeq L(x, y)$  where  $f : \mathbb{R} \mapsto \mathbb{R}$  is the camera response function and  $r : (x, y) \subset \mathbb{R}^2 \mapsto \mathbb{R}$  is the image irradiance function. Below we will represent the image intensity function with its corresponding scale-

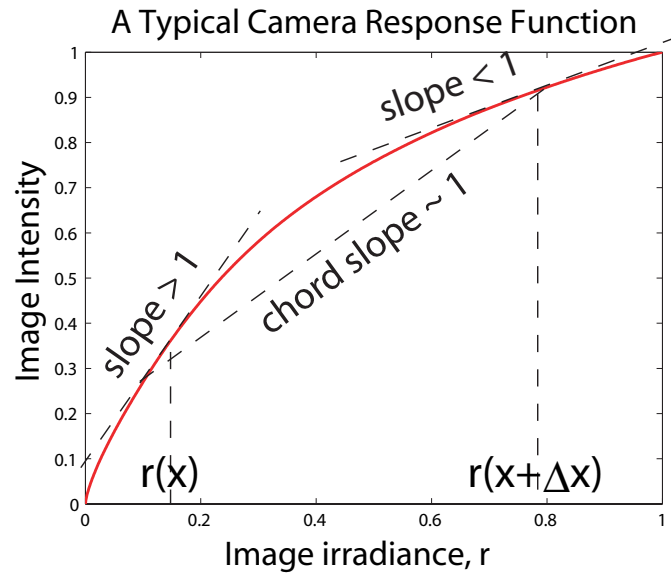


Figure 3.4: A typical concave camera response function.  $M$  is the image irradiance function

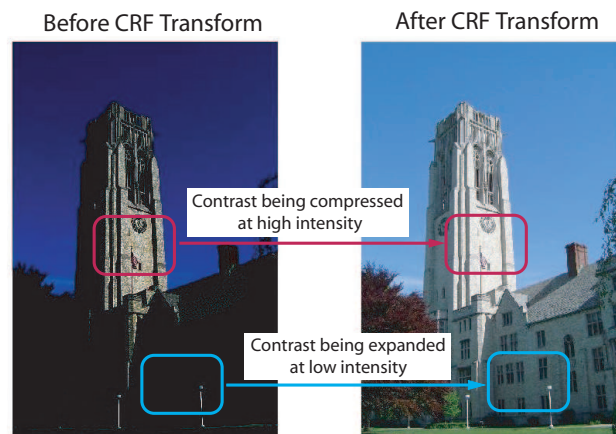


Figure 3.5: A visual effect of camera response function transform at compressing contrast at high intensity values and expanding contrast at the low intensity values.

space function  $L(x, y)$ . By the chain rule, we have:

$$L_x = \frac{\partial L}{\partial x} = \frac{df}{dr} \frac{\partial r}{\partial x}, \quad L_y = \frac{\partial L}{\partial y} = \frac{df}{dr} \frac{\partial r}{\partial y} \quad (3.12)$$

Note that the modulation factor,  $\frac{df}{dr}$  is the derivative of the camera transfer function, which is larger (smaller) than 1 when  $r$  is small (large). Therefore, after the transformation, the Euclidean gradient  $|\nabla L| = \sqrt{L_x^2 + L_y^2}$  of a transformed image is higher (lower) at the low (high) intensity than before. Namely, the modulation term  $\frac{df}{dr}$  in Equ. (3.12) reveals a key difference between PIM and PRCG, when PRCG images are not subjected to such modulation on their gradient values. If the PRCG intensity functions have not undergone such transformation, it can be distinguished from PIM by the gradient distribution.

The analysis above assumes that the image irradiance function  $r$  is continuous. There is a non-trivial issue involved in its implementation, when it comes to discrete-sampled images. Consider approximating the gradient at two neighboring pixels at locations  $x$  and  $x + \Delta x$ , Equ. (3.12) becomes:

$$\frac{\Delta L_x}{\Delta x} = \frac{\Delta(f \circ r)_x}{\Delta r_x} \frac{\Delta r_x}{\Delta x} \quad (3.13)$$

where  $\Delta L_x$  represents  $L(x + \Delta x, y) - L(x, y)$ , and similar representation is applied for  $\Delta r_x$  and  $\Delta(f \circ r)_x$  in Fig. 3.4. Note that, the modulation factor in this case becomes the slope of the chord on the camera response function connecting  $r(x + \Delta x)$  to  $r(x)$ . Therefore, the modulation factor will only be similar to that of the continuous case, when  $|r(x + \Delta x) - r(x)|$  is small, otherwise the slope of the chord would be approaching 1 and modulation effect becomes weak. Where  $|r(x + \Delta x) - r(x)|$  is small,  $|\frac{\Delta r_x}{\Delta x}|$  would be small too. As a result, due to the discrete image representation,

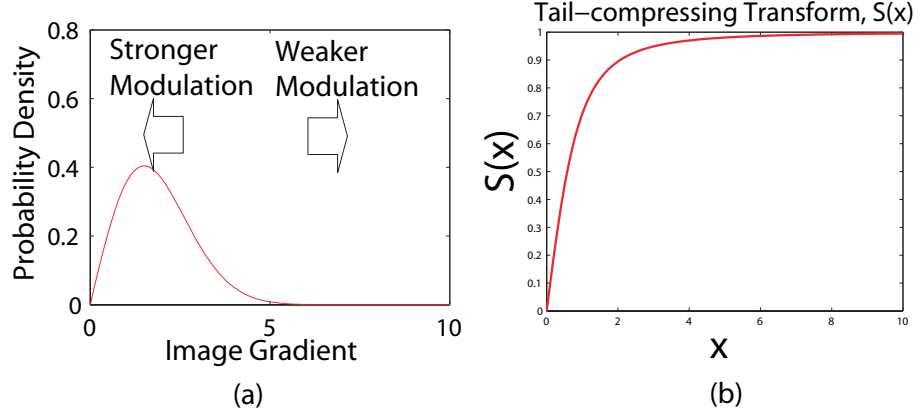


Figure 3.6: (a) The relative strength of the modulation effect for different magnitude of the image irradiance gradient (b) The tail-compressing function,  $S(x; 1)$

the modulation effect shown in Equ. (3.13) would be more prominent at points with low gradient values. This idea is illustrated in Fig. 3.6 (a). Although our analysis is based on a simple derivative estimation model in Equ. 3.13, the analysis result holds qualitatively in general even when we use a more complicated method to estimate the image derivatives, because a large irradiance value gap between two adjacent pixels could not capture the detailed CRF shape between the irradiance values. With this property, if we are to compare the distributions of the gradient of PIM with that of PRCG, we should place more weight on the low gradient region than on the high gradient region. Fig. 3.6 (a) shows a sample distribution of gradient values of an image, which typically contains long tail. To emphasize the low-gradient region, we employ a tail-compressing transform  $S$  as shown in Fig. 3.6 (b):

$$S(|\nabla L|; \alpha) = \sqrt{\frac{|\nabla L|^2}{\alpha^{-2} + |\nabla L|^2}} = |\text{grad}(\alpha L)|, \quad \text{where } |\nabla L| = \sqrt{L_x^2 + L_y^2} \quad (3.14)$$

Fig. 3.6 (b) shows that the  $S$  transform is almost linear for small values and that it compresses high values. The width of the linear range can be controlled by the

constant,  $\alpha$ . Interestingly, Equ. (3.14) is the surface gradient of the scaled image intensity function  $|\text{grad}(\alpha L)|$  computed from the Riemannian metric for the graph of a single channel intensity function (see Appendix C for derivation).

Fig. 3.7 shows the distribution of the mean of surface gradient  $|\text{grad}(\alpha L)|$ ,  $\alpha = 0.25$  (selected such that the linear range of the  $S$  transform covers the low and the intermediate value for Euclidean gradient), for three intensity ranges, i.e.,  $[0, \frac{1}{3})$ ,  $[\frac{1}{3}, \frac{2}{3})$  and  $[\frac{2}{3}, 1]$ , of the blue-color channel (the same holds for the red and green channels). These distributions are computed empirically from our actual dataset of PIM and PRCG. Notice that for the low intensity region, the mean of surface gradient for the PIM is higher than that of the PRCG and the opposite is observed for the high intensity region, while the distributions of the two are completely overlapped at the medium intensity range. This perfectly matches our prediction about the effect of the transfer function as described earlier in Equ. (3.12).

For feature extraction, we compute the surface gradient and obtain  $(|\text{grad}(\alpha L^R)|, |\text{grad}(\alpha L^G)|, |\text{grad}(\alpha L^B)|)$ ,  $\alpha = 0.25$  for the three color channels at every pixel of an image. As CRF in PIM modulates the gradient differently at different intensity values, we combine the intensity values  $(L^R, L^G, L^B)$  with the above surface gradient vector at every pixel, and form a vector field  $(L^R, L^G, L^B, |\text{grad}(\alpha L^R)|, |\text{grad}(\alpha L^G)|, |\text{grad}(\alpha L^B)|)$  on the  $F^{RGB}$  submanifold. In Sec. 3.9, we will show how features are extracted for classification by computing the rotational moments of the 6D vectors.

### 3.7.2 The Second Fundamental Form

Referring to the **Object Model Difference** as mentioned in Section 3.3, the accuracy of the 3D polygonal model of computer graphics is dependent on the granularity of the polygonal representation. A coarse-grained polygonal model can result

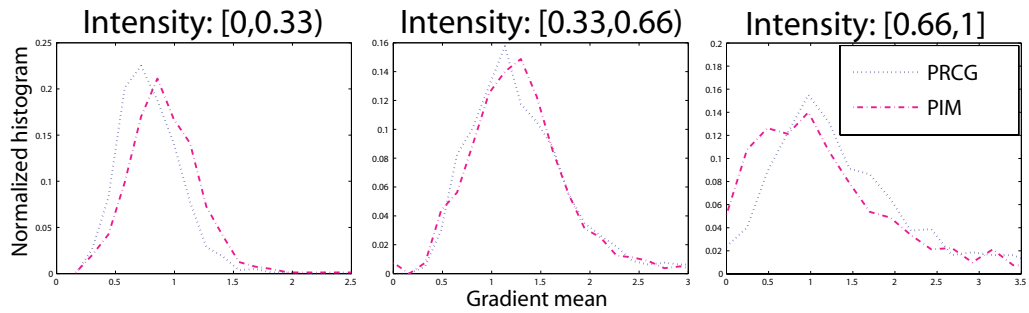


Figure 3.7: Distribution of the surface gradient for three different intensity ranges of the blue color channel. Each of the distribution is respectively computed from the entire image set

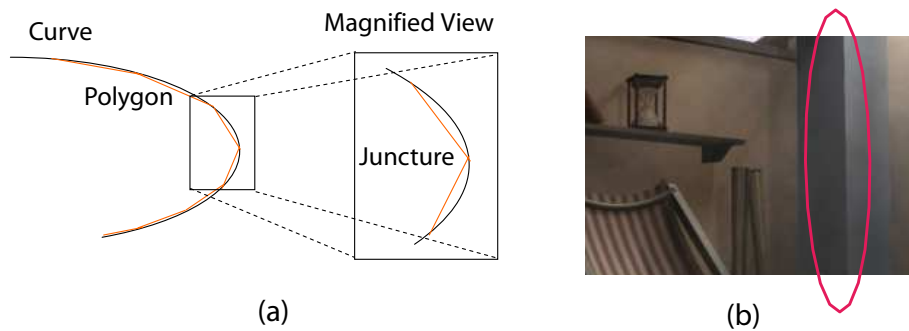


Figure 3.8: (a) Illustration of the polygon effect. (b) Unusually sharp structure in computer graphics, note the pillar marked by the red ellipse.

in observable sharp structures, such as sharp edges and sharp corners, in the image intensity function of computer graphics. Fig. 3.8(a) gives an intuitive illustration of this point; when a concave line is approximated by a polygonal line, the curvature at the junctures of the polygon segments is always greater than that of the concave line. Fig. 3.8(b) shows an example of the sharp edge structure in the magnified view of a PRCG image. This structural difference between PIM and PRCG can be observed from the local quadratic geometry computed on the image intensity function. Quadratic geometry can be considered as a second-order approximation of the intensity function surface at a point. The typical shapes of quadratic geometry are shown in Fig. 3.9(a).

In differential geometry, the quadratic geometry at each point  $(x, y)$  of an image intensity function graph  $F^I(x, y)$  is related to the second fundamental form. Note that,  $F^I(x, y)$  is a submanifold of co-dimension one, which is also known as a hyper-surface. The co-dimension of a submanifold is defined as difference between the dimension of the ambient space and that of the submanifold. In [44], the second fundamental form at a point  $p$  of a hyper-surface  $M$  is defined on the tangent plane  $T_p M$  as below:

$$\Pi_p(V) = \langle AV, V \rangle_g = \langle V, AV \rangle_g, \quad \text{where } V \in T_p M \quad (3.15)$$

which can also be written as the form of a quadratic function as in Equ 3.16 when the tangent vector  $V$  is represented in orthonormal basis.

$$\Pi_p(V) = V^T AV \quad (3.16)$$

For the definition in Equ. 3.15,  $A$  is the Weingarten map or the shape operator of

$M$ , which is a self-adjoint linear transformation on  $T_pM$  (i.e.,  $\langle AV, V \rangle_g = \langle V, AV \rangle_g$ ). From Equ 3.16, matrix  $A$  can be thought as the Hessian of the local surface, when it is locally represented as a graph over its tangent plane. Therefore,  $A$  determines the local quadratic geometry of  $M$ , which can be characterized by the two eigenvalues of  $A$ ,  $\gamma^1$  and  $\gamma^2$ , with  $\gamma^1 > \gamma^2$ . The eigenvalues are called the local principal curvatures of  $M$ .

For  $F^I(x, y) = (x, y, L(x, y))$ , the unit normal vector  $n(x, y)$  is given by:

$$n(x, y) = \frac{(-L_x, -L_y, 1)}{\sqrt{1 + L_x^2 + L_y^2}} \quad (3.17)$$

Then,  $A$  is given by:

$$\begin{aligned} A &= \begin{pmatrix} \langle F_{xx}^I, n \rangle & \langle F_{xy}^I, n \rangle \\ \langle F_{xy}^I, n \rangle & \langle F_{yy}^I, n \rangle \end{pmatrix} \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix}^{-1} \\ &= \frac{1}{(1 + L_x^2 + L_y^2)^{\frac{3}{2}}} \begin{pmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{pmatrix} \begin{pmatrix} L_y^2 + 1 & -L_x L_y \\ -L_x L_y & L_x^2 + 1 \end{pmatrix} \\ &= \frac{1}{(1 + L_x^2 + L_y^2)^{\frac{3}{2}}} \begin{pmatrix} L_{xx}(L_y^2 + 1) - L_{xy}L_xL_y & L_{xy}(L_x^2 + 1) - L_{xx}L_xL_y \\ L_{xy}(L_y^2 + 1) - L_{yy}L_xL_y & L_{yy}(L_x^2 + 1) - L_{xy}L_xL_y \end{pmatrix} \\ &= \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \end{aligned} \quad (3.18)$$

With  $A$  given by Equ. 3.18, its first and its second eigenvalues can be computed as:

$$\{\gamma^1, \gamma^2\} = \frac{-(a_{11} + a_{22}) \pm \sqrt{(a_{11} - a_{22})^2 - 4a_{21}a_{12}}}{2}, \quad \gamma^1 > \gamma^2 \quad (3.19)$$



In a 2D plot of the first and the second eigenvalues, every point represents a local quadratic geometry shape, as shown in Fig. 3.9(b) (The meaning of the circles, ellipses and so on is given in Fig. 3.9(a)). Note that, large eigenvalues correspond to the ‘sharp’ structures such as sharp ellipses or sharp circles. Given an image, the presence of the large eigenvalues can be measured by the skewness of the distribution of the eigenvalues in each image (Skewness may not be the best measure, but it serves our purpose for illustration); the larger the skewness is, the more large values are there. We compute the skewness of the eigenvalues separately for the images in our dataset and the distribution of the skewness is shown in Fig. 3.10. We can see that the *CG* image set tends to have a larger skewness, while the shape of the distributions for the two photographic sets (*Google* and *Personal*) are quite consistent. This observation indicates that PRCG has more sharp structures than PIM.

For feature extraction, we compute the two eigenvalues of the quadratic form for the three-color channels independently. As the geometry of the edge region and the non-edge regions are different in terms of the generative process, (e.g., low-gradient region is mainly due to smooth surface while high-gradient region is mainly due to texture, occlusion, change of the surface reflectance property or shadow), we therefore try to capture the correlation between image gradient and the local quadratic geometry with a combined vector  $(|\nabla L^R|, |\nabla L^G|, |\nabla L^B|, \gamma_R^1, \gamma_G^1, \gamma_B^1, \gamma_R^2, \gamma_G^2, \gamma_B^2)$  at every pixel, and hence it forms a vector field on the  $F^{RGB}$  submanifold. In Sec. 3.9, we will show how features are extracted for classification by computing the rotational moments of the 9D vectors.

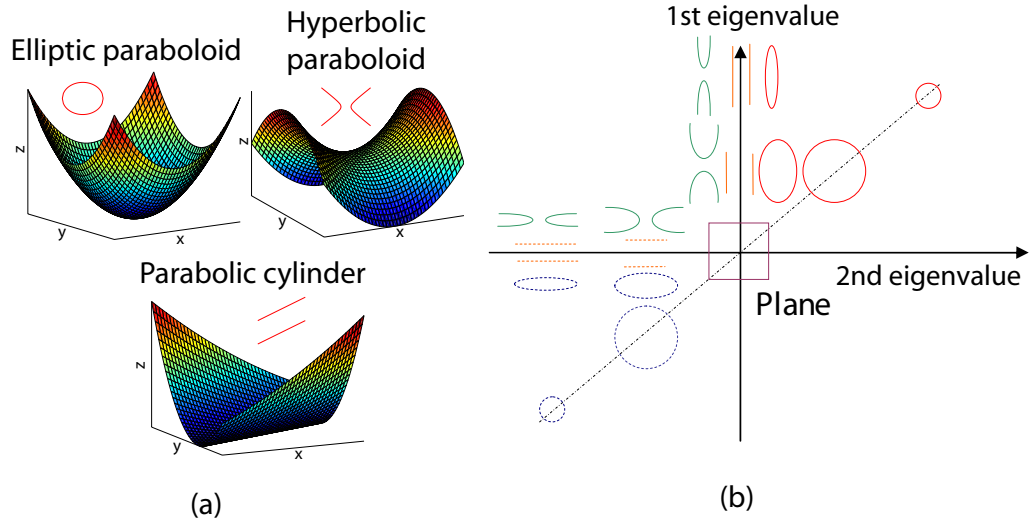


Figure 3.9: (a) The typical shapes of the quadratic geometry (b) The shapes of the quadratic geometry in a 2D eigenvalue plot. Colors are for visual aid

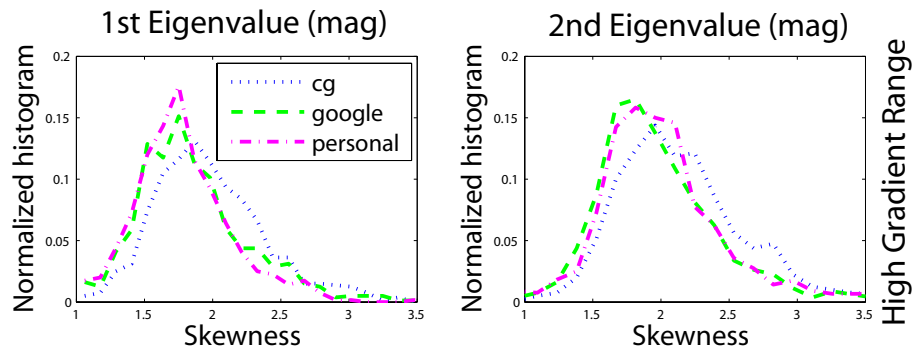


Figure 3.10: Distribution of the skewness of the 1st and 2nd eigenvalues of the second fundamental form for the blue color channel

### 3.7.3 The Beltrami Flow Vector

In Section 3.3, we discussed the **Surface Model Difference** between PIM and PRCG. The object reflectance property function in PRCG is often simplified such that its response to different color channels or spectral components are independent. This assumption is not true for real-world objects in general, therefore this simplification may result in a deviation of the cross-color-channels relationship of PRCG with respect to that of PIM. Such joint-color-channel correlation has been used by some techniques in image restoration to improve the subjective image quality. Therefore, we consider the Beltrami flow [91], which is an effective joint-color-channel image restoration technique. Beltrami flow is based on the idea of minimizing the local surface area, which has been employed for restoring degraded or noisy images where artifacts or noise are considered as singularities on the image intensity function. Minimization of the surface area reduces the magnitude of the singularity.

For the graph of a joint-RGB image intensity function

$$F^{RGB}(x, y) = (x, y, L^R(x, y), L^G(x, y), L^B(x, y))$$

Beltrami flow [91] is defined as a flow partial differential equation (PDE):

$$L_t^i = \Delta_g L^i \tag{3.20}$$

where  $i \in R, G, B$ . In Equ. 3.20,  $\Delta_g$  is the Beltrami operator, and  $\Delta_g L^i$  is the Beltrami flow vector. From the flow PDE, the graph manifold  $F^{RGB}$  changes continuously according to the Beltrami flow vector  $\Delta_g L^i$ , which is defined as:

$$\Delta_g L^i = \frac{1}{\sqrt{|g|}} (\partial_x (\sqrt{|g|} (g^{xx} L_x^i + g^{xy} L_y^i)) + \partial_y (\sqrt{|g|} (g^{yx} L_x^i + g^{yy} L_y^i))) \tag{3.21}$$

In Equ. 3.21,  $|g|$  is determinant of matrix  $g$  which can be written as:

$$|g| = g_{xx}g_{xy} - g_{xy}^2 = 1 + \sum_j |\nabla(L^j)|^2 + \frac{1}{2} \sum_{l,k} |\nabla(L^l) \times \nabla(L^k)|^2, \quad l, j, k = R, G, B \quad (3.22)$$

and  $g^{ij}$  is the elements of  $g^{-1}$ , which is given by:

$$g^{-1} = \frac{1}{g_{xx}g_{xy} - g_{xy}^2} \begin{pmatrix} g_{yy} & -g_{xy} \\ -g_{yx} & g_{xx} \end{pmatrix} \quad (3.23)$$

In Equ. 3.22  $\nabla(L^l)$  and  $\nabla(L^k)$  are both 2D vectors, but they will be extended to vectors in  $\mathbb{R}^3$  before computing the vector cross-product terms  $\nabla(L^l) \times \nabla(L^k)$ . Note that the vector cross-product terms in Equ. 3.22 capture the correlation of the gradients in the R, G and B color channels. We can visualize the 3D joint distribution of the Beltrami flow vectors from the 2D plots of  $\Delta_g L^R - \Delta_g L^G$  and  $\Delta_g L^R - \Delta_g L^B$ . For the 2D plots of  $\Delta_g L^R - \Delta_g L^G$  and  $\Delta_g L^R - \Delta_g L^B$ , we notice that the distribution of the PIM is more aligned to the  $y = x$  line of the plots, while the PRCG tends to have misaligned points or outliers. This observation can be seen in Fig. 3.11, showing the 2D plots of a PRCG together with those of its recaptured counterpart. Note that the PRCG and its recaptured counterpart have the same content, and latter is a PIM (acquired by a camera). We visually inspected 100 *CG* images and 100 *Google* images, and noticed that about 20% of the *CG* images have this misalignment as compared to less than 5% of the *Google* images. Such misalignment could be due to the spectral independence assumption for the surface reflectance function.

For feature extraction, we follow the same strategy as the second fundamental form and try to capture the correlation between the Euclidean gradient and the

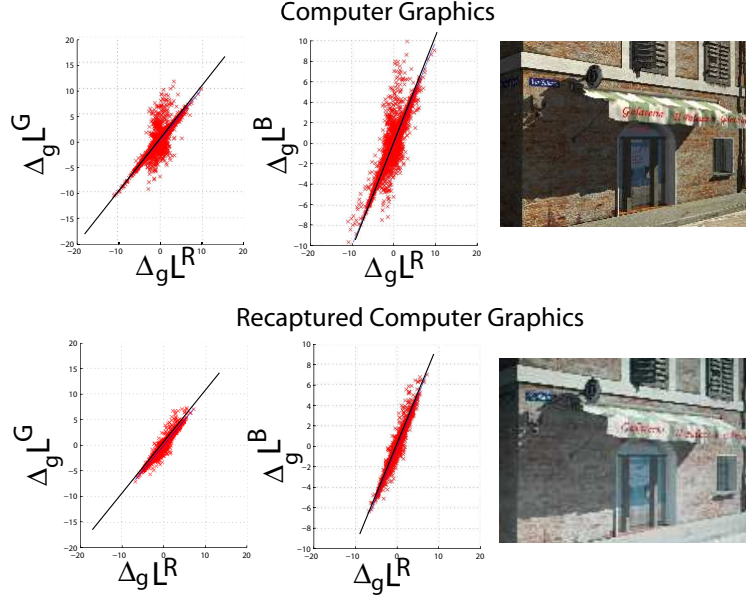


Figure 3.11: Comparing the joint distribution of the Beltrami flow components of a computer graphics image and that of its recaptured counterpart, the lines correspond to  $y = x$

Beltrami flow vector. As such, we compute  $(|\nabla L^R|, |\nabla L^G|, |\nabla L^B|, \Delta_g L^R, \Delta_g L^G, \Delta_g L^B)$  at every pixel, which forms a vector field on the  $F^{RGB}$  submanifold. In Sec. 3.9, we will show how features are extracted for classification by computing the rotational moments of the 6D vectors.

### 3.7.4 Normalization of Differential Geometry Features

The differential-geometric quantities are not invariant to image scaling. In this subsection, we propose a normalization scheme to reduce their sensitivities to image scaling.

When we compute the differential-geometric quantities, we are essentially computing the derivatives of various orders in scale-space at a fixed scale  $s = 1$  pixel. We like to make sure that the geometric quantities are as invariant as possible when we resize an image. To understand the effect of these operations on the scale-space

computation, consider a simple example. For  $f_1(x) = \cos(\omega x)$ , the scale-space first derivative (the conclusion can be easily generalized to the higher-order derivatives) is given by:

$$L_{1x}(x; s) = -\omega e^{-\frac{\omega^2 s}{2}} \sin(\omega x) \quad (3.24)$$

Let's resize  $f_1(x)$  by a factor  $k$  and obtain  $f_k(x) = \cos(k\omega x)$ ; the corresponding scale-space first derivative would be:

$$L_{kx}(x; s) = -k\omega e^{-\frac{k^2\omega^2 s}{2}} \sin(k\omega x) \quad (3.25)$$

As  $f_1(x) = f_k(\frac{x}{k})$ , we compare  $L_{1x}(x; s)$  in Equ. (3.24) with  $L_{kx}(\frac{x}{k}; s) = -k\omega e^{-\frac{k^2\omega^2 s}{2}} \sin(\omega x)$  in order to understand the effect of image resizing. The difference between  $L_{1x}(x; s)$  and  $L_{kx}(\frac{x}{k}; s)$  is on the preceding factor  $k$  and the exponential factor. The first factor represents a systematic effect and is independent of the original signal, whereas the exponential factor, being a function of  $\omega$ , is content dependent. We propose a way to minimize the effect of image resizing by removing the systematic effect of the first factor. Note that this observation is also applicable to the general differential-geometric quantities which are composed of scale-space derivatives. If we compute a differential-geometric quantity at every pixel of an image and obtain the distribution of this quantity for each image, the mentioned first factor will manifest itself in the standard deviation of the distribution. Therefore, we propose a simple divisive normalization scheme, that divides the differential geometry quantity of an image by its standard deviation. To prove the effectiveness of such a normalization, we compute the Kullback-Leibler (KL) distance [11] between the distribution of the scale-space first derivative of an image and that of the half-sized version of the same image. Indeed, we find that the KL distance is reduced to about one-third after

normalization.

Apart from image scaling, image rotation is also a common image operation. All of our features are rotation-invariant, except for the Beltrami flow vector, and the local patch statistics feature which we will describe next.

### 3.8 Local Patch Statistics

Natural image statistics [92] represents the statistical regularities inherent in natural images (defined as images commonly encountered by human). Natural image statistics can be applied as an image prior for applications such as image compression, image recognition and image restoration. The important image statistics are the power law of the natural-image power spectrum [22], the wavelet high-kurtotic marginal distribution, and the higher-order cross-subband correlation of the wavelets coefficients [88]. The wavelet features proposed in [55] are derived from these wavelet-based natural image statistics.

In addition to the transform-domain image statistics, an image-space natural image statistic was proposed [43]: The authors studied the high-dimensional distribution of  $3 \times 3$  high-contrast local patches which mainly correspond to the edge structures. They found that the distribution is highly structured and concentrates on a 2D manifold in an 8D Euclidean space. By using this method, they managed to uncover the statistical difference between the optical (camera) images and the range (laser scan) images. Just like the PIM and the PRCG, these two groups of images correspond to two distinct physical image generation processes. There is further evidence [83] that local patches can actually capture image styles, where painting, line drawing, computer graphics, photographs and even images of different resolutions can be considered as having different styles. The local patch model has been suc-

cessfully applied to demonstrate image style translation [83], super-resolution [24], and other applications. This motivates us to employ local patch statistics.

Now, we describe the procedure for extracting the local patch statistic features. We extract (see Fig. 3.12(a) & (b)) the grayscale patch and the joint-spatial-color patch independently at the edge points in two types of image regions: the high contrast region, and the low but non-zero contrast region. The two regions are obtained by thresholding the  $D$ -norm defined below. Note that the joint-spatial-color patch is approximately oriented to the image gradient direction which is the direction of maximum intensity variation. Each sampled patch, represented as a vector  $\mathbf{x} = [x_1, x_2, \dots, x_9]$ , is mean-subtracted and contrast-normalized as in Equ. (3.26):

$$\mathbf{y} = \frac{\mathbf{x} - \bar{x}}{\|\mathbf{x} - \bar{x}\|_D} \quad (3.26)$$

where  $\bar{x} = \frac{1}{9} \sum_{i=1}^9 x_i$  and  $\|\cdot\|_D$  is called  $D$ -norm.  $D$ -norm is defined as  $\|\mathbf{x}\|_D = \sqrt{\sum_{i \sim j} (x_i - x_j)^2}$  with  $x_i$  and  $x_j$  representing the patch elements and  $i \sim j$  denoting the 4-connected neighbors relationship of two pixels in a patch. The  $D$ -norm can also be expressed as the square root of a quadratic form  $\|\mathbf{x}\|_D = \sqrt{\mathbf{x}^T D \mathbf{x}}$  where  $D$  is symmetric semi-positive definite matrix [43].

As the patch  $\mathbf{x}$  is contrast-normalized by the  $D$ -norm, the normalized patch is constrained by a quadratic relationship,  $\mathbf{y}^T D \mathbf{y} = 1$ , which implies that the data points are living on the surface of an ellipsoid in 9D Euclidean space. To facilitate the handling of the data points in a high-dimensional space, the elliptic constraint can be transformed into a spherical constraint by a linear transformation  $\mathbf{v} = M \mathbf{y}$ , where  $M$  is a  $8 \times 9$  matrix and the resulting  $\mathbf{v}$  is constrained by  $\mathbf{v}^T \mathbf{v} = \|\mathbf{v}\|^2 = 1$ , which implies that  $\mathbf{v}$  is located on 7-sphere in a 8D Euclidean space, as illustrated in Fig. 3.12(c) in a 3D example. In this process,  $\mathbf{v}$  is reduced from 9D to 8D by taking advantage



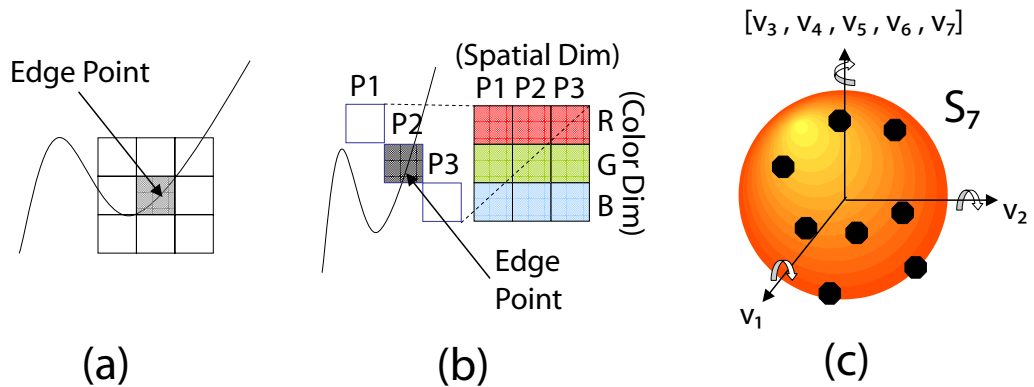


Figure 3.12: The grayscale patch (a) and the joint-spatial-color patch (b) are sampled at the edge points in an image. (c) Point masses on  $S^7$ , a 7D sphere.

of the fact that  $\mathbf{y}$  is zero-mean. For the each of the four patch types (grayscale/low-contrast, grayscale/high-contrast, joint-spatial-color/low-contrast, and joint-spatial-color/high-contrast), we extract 3000 patches and separately construct a distribution on a 7-sphere in the 8D Euclidean space.

### 3.9 Description of Joint Distribution by Rigid Body Moments

We have described several distributions of features such as the 8D local patch feature and the differential geometry quantities computed for every pixel in an image. Here we propose to extract the statistics of these distributions in order to reduce the dimensionality and develop a classification model for distinguishing PIM and PRCG.

There are many ways to describe this high-dimensional distribution. If the data points lie on the surface of a sphere like the case of the local patch feature, one way would be to uniformly quantize the surface of the sphere into 17520 bins (or other suitable bin count) to form a 1D histogram [43]. This method needs a large number of image patches in order to have a stable histogram if the distribution is relatively

spread out. In the other extreme, a non-parametric density estimation method, such as the Parzen kernel based method, can be used but the computation cost would be intensive when it comes to computing the distance between two estimated density functions. Besides that, Gaussian mixture model (GMM) clustering can also be used, but the standard GMM algorithm does not take advantage of the (non-Euclidean) spherical data space.

Due to the above considerations and the concern about the computational cost, we develop a framework based on the rigid body moments, which is capable of handling a high-dimensional distribution and is especially suitable for a spherical distribution. Let's first describe the process of computing the rotational rigid-body moments for the local patch distribution in the form of a discrete set of point masses on a 7-sphere [62]. For a distribution of  $N$  discrete masses,  $m_i$ ,  $i = 1, \dots, N$ , respectively located at  $\mathbf{v}_i = [v_{i1} \dots v_{iM}]$ , the element of the inertia matrix is given by (3.27),

$$I_{jk} = \sum_i^N m_i (\|\mathbf{v}_i\|^2 \delta_{jk} - v_{ij} v_{ik}) \quad j, k = 1, \dots, M \quad (3.27)$$

where the Kronecker delta  $\delta_{ij}$  is defined as 0 when  $i \neq j$ , 1 when  $i = j$ . For an example of a 3D Euclidean space of  $x$ - $y$ - $z$  Cartesian coordinates, the inertia matrix would be:

$$I = \sum_i^N m_i \begin{pmatrix} y_i^2 + x_i^2 & -x_i y_i & -x_i z_i \\ -x_i y_i & z_i^2 + x_i^2 & -y_i z_i \\ -x_i z_i & -y_i z_i & x_i^2 + y_i^2 \end{pmatrix} \quad (3.28)$$

Notice that the inertia matrix is symmetric. The diagonal and the off-diagonal components are respectively called the moments of inertia and the products of inertia. For an  $n$ -dimensional space, the inertia matrix has  $n$  moments of inertia and  $\frac{1}{2}n(n-1)$  unique products of inertia.

For the distribution on the 7-sphere with  $N$  data points, we assign the mass for each data point as  $\frac{1}{N}$ . We extract only the moment of inertia, as the number of the unique products of inertia is large. From another perspective, we can consider the elements of the inertia matrix as the second-order statistics of the distribution, and therefore it does not capture the complete information of distribution. Besides the moments of inertia, we also compute the center of mass (a form of the first-order statistics) as well as the mean and the variance of the distance of the data points from the center of mass.

However, the feature vectors for the fractal dimension, the surface gradient, the second fundamental form and the Beltrami flow vector are not confined to a unit sphere. In this case, the inertia quantities can be affected by two factors: the distribution of points in the spherical direction and that in the radial direction. We decouple the two factors and model their effects separately: we model the distribution of the normalized unit-length data vectors (which lie on a unit sphere) using the center of mass as well as the moments and the products of inertia, and model the magnitude of the data vectors using the first four moments of the distribution, i.e., mean, variance, skewness and kurtosis.

Fig. 3.13 shows the feature distribution of the four local patch types, after having been linearly projected to a 2D space by Fisher discriminant analysis. The ellipses in the figure depict the mean and the covariance of a single-class feature distribution; a more separable pair of ellipses indicates that the two corresponding distributions are more separable. We observe that the joint-spatial-color patch provides a better discrimination between the PIM and PRCG. Fig. 3.14 shows the same 2D linear projection of the fractal, the surface gradient, the 2nd fundamental form and the Beltrami flow vector feature distribution. Notice that fractal dimension feature is the weakest discriminant for PIM and PRCG and differential-geometric features are

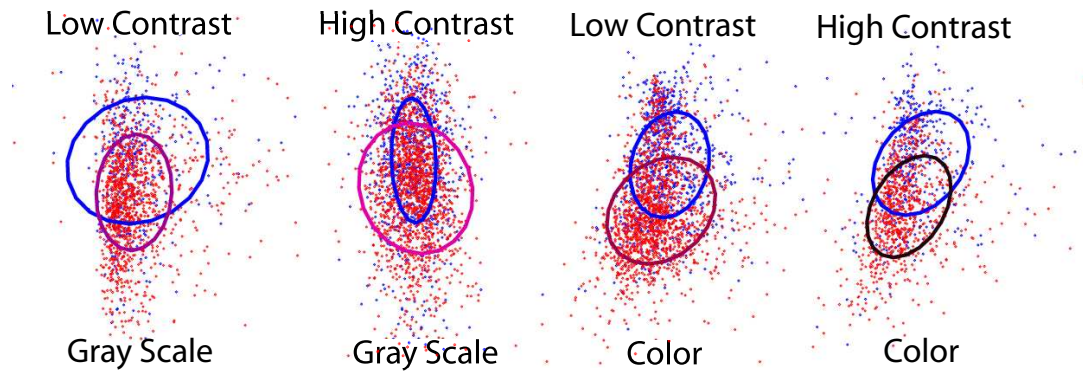


Figure 3.13: 2D projection of the local patch feature distribution for the (*Google+personal* image sets (red) and the *CG* image set (blue)

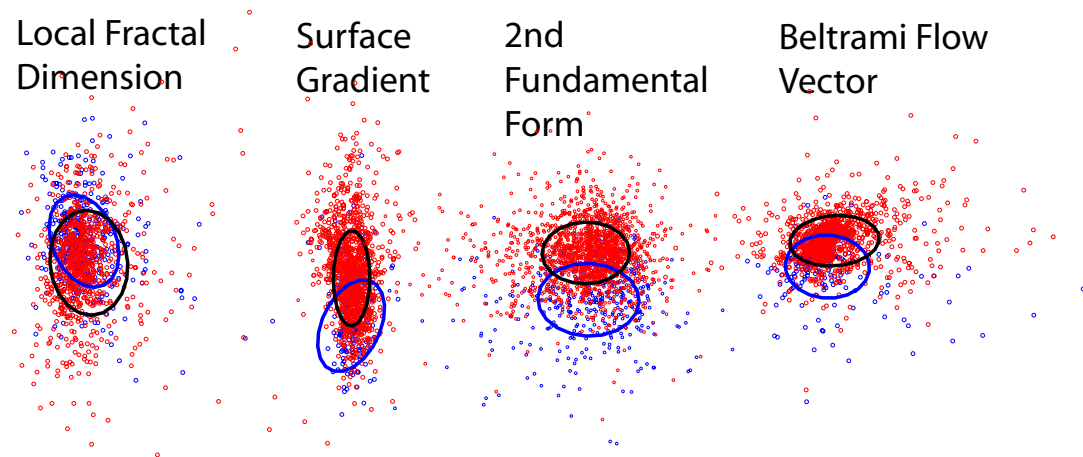


Figure 3.14: 2D projection of the fractal, the surface gradient, the 2nd fundamental form and the Beltrami flow vector features (from left to right) for the (*Google+personal* image sets) (red) and the *CG* image set (blue)

strong discriminant features.

### 3.10 Columbia Photographic Images and Photorealistic Computer Graphics Dataset

To ensure that our experimental dataset exemplifies the problem of image forgery detection, our dataset collection effort adheres to the following principles: (1) Images of diverse but natural-scene-only content: we exclude the PRCG of fantasy or

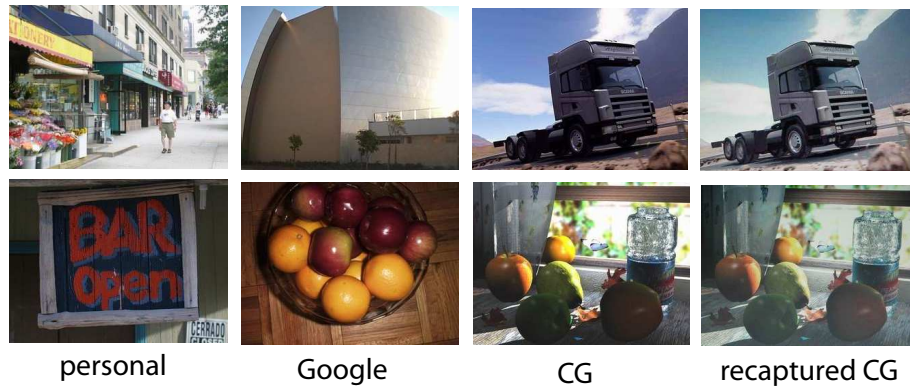


Figure 3.15: Examples from our image sets. Note the photorealism of all images.

abstract category and this restriction ensures a content matching between the PIM and the PRCG image sets, (2) Computer graphics of high photorealism: we subjectively filter the computer graphics from the web to ensure a high photorealism, (3) Images of reliable ground truth: we specifically collect a set of PIM from the personal collections which are known to be authentic. As a comparison, a very different approach in terms of the dataset is adopted in [55], where a very large number of online images, i.e., 32,000 PIM and 4,800 PRCG, are used and the selection criteria is not given. Adhering to the above principles, we collected the below-described four image sets (see Fig. 3.15). A detailed description of the dataset can be found in [69].

1. 800 PRCG (*CG*): These images are categorized by content into architecture, game, nature, object and life, see Fig. 3.16(a). The PRCG are mainly collected from various 3D artists (more than 100) and about 40 3D-graphics websites, such as [www.softimage.com](http://www.softimage.com), [www.3ddart.org](http://www.3ddart.org), [www.3d-ring.com](http://www.3d-ring.com) and so on. The rendering software used are such as 3ds MAX, Softimage-XSI, Maya, Terragen and so on. The geometry modelling tools used include AutoCAD, Rhinoceros, Softimage-3D and so on. High-end rendering techniques used



(a) Computer Graphics



(b) Author's Personal

Figure 3.16: (a) Subcategories within *CG* and (b) Subcategories within *personal* image set, the number is the image count.

include global illumination with ray tracing or radiosity, simulation of the camera depth-of-field effect, soft-shadow, caustics effect and so on.

2. 800 PIM images (*personal*): 400 of them are from the personal collection of Philip Greenspun, they are mainly travel images with content such as indoor, outdoor, people, objects, building and so on. The other 400 are acquired by the authors using the professional single-len-reflex (SLR) Canon 10D and Nikon D70. It has content diversity in terms of indoor or outdoor scenes, natural or artificial objects, and lighting conditions of day time, dusk or night time. See Fig. 3.16(b).
3. 800 PIM from Google Image Search (*Google*): These images are the search results based on keywords that matches the content types of images seen in the computer graphics category described above. The keywords are such as architecture, people, scenery, indoor, forest, statue and so on.
4. 800 photographed PRCG (*recaptured CG*): These are the photograph of the screen display of the mentioned 800 computer graphics. Computer graphics are displayed on a 17-inch (gamma linearized) LCD monitor screen with a display resolution of 1280×1024 and photographed by a Canon G3 digital camera. The acquisition is conducted in a dark room in order to reduce the reflections from the ambient scene.

The rationale of collecting two different sets of PIM is the following: *Google* has a diverse image content and involves more types of cameras, photographer styles and lighting conditions but the ground truth may not be reliable, whereas *personal* is reliable in terms of the ground truth but it is limited in the camera and photographer style factors. On the other hand, we produce the *recaptured CG* image set by



recapturing the PRCG using a camera, so that we can evaluate how well recapturing PRCG will escape the PRCG detector, as a form of attack on the PRCG detector. This dataset is open to the research community with the name the *Columbia Photographic Images and Photorealistic Computer Graphics Dataset* and available at <http://www.ee.columbia.edu/trustfoto>.

Different image sets have different average resolution. To prevent the classifier from learning the resulting content-scale difference, we resize the *personal* and *re-captured CG* sets, such that the mean of the averaged dimension,  $\frac{1}{2}(\text{height} + \text{width})$  of the image sets matches that of the *Google* and the *CG* sets, at about 650 pixels.

### 3.11 Experiments

We evaluate the capability of our geometry-based features (henceforth the geometry feature) by classification experiments on our image sets. We compare the 192D geometry feature against the 216D wavelet feature [55] and the 108D feature obtained from modelling the characteristics of the general (i.e., including both photorealistic and non-photorealistic) computer graphics [39] (henceforth the cartoon feature). For a fair comparison, we compute the wavelet feature on the entire image (for a better performance), rather than just on the central  $256 \times 256$ -pixel region of an image, as described in [55]. The cartoon feature consists of the average color saturation, the ratio of image pixels with brightness greater than a threshold, the Hue-Saturation-Value (HSV) color histogram, the edge orientation and strength histogram, the compression ratio and the pattern spectrum.

The classification experiment is based on the Support Vector Machine (SVM) classifier of the LIBSVM [37] implementation. We use the radial basis function (RBF) kernel for the SVM and model selection (for the regularization and the kernel



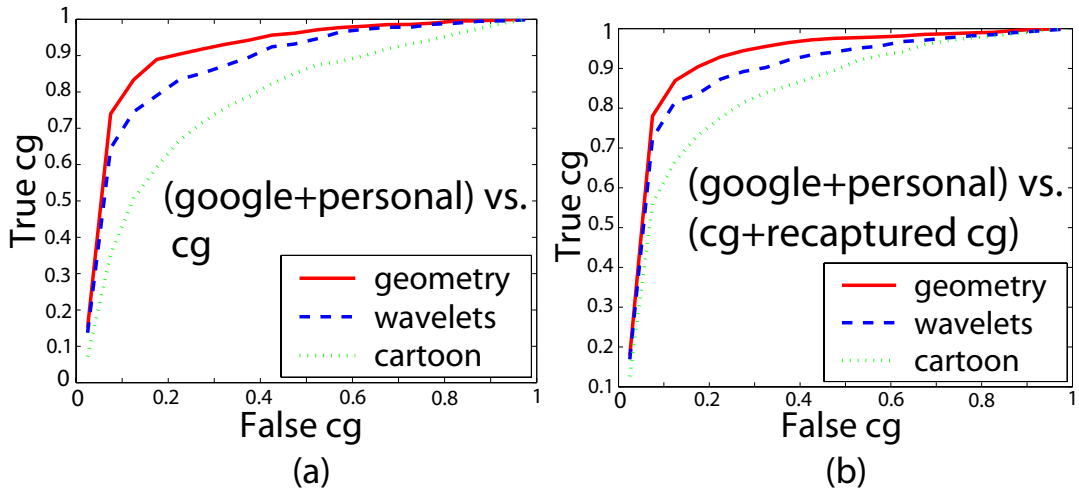


Figure 3.17: Receiver operating characteristic (ROC) curve for two classification experiments

Table 3.1: Classifier Test Accuracy

Set	Test images (count)	Wavelets	Geometry
A	recaptured CG (800)	<b>97.2%</b>	96.6%
B	photos of artificial objects (142)	94.0%	<b>96.2%</b>
C	CG of nature scenes (181)	<b>57.5%</b>	49.2%
D	CG of living objects (50)	64.0%	<b>74.0%</b>
E	CG with DOF simulation (21)	85.7%	<b>90.5%</b>

parameters) is done by a grid search [37] in the joint parameter space. The classification performance we report hereon is based on a five-fold cross-validation procedure. We train a classifier of the PIM (*Google+personal*) versus the PRCG (*CG*), based on the geometry, wavelet and cartoon features respectively. The receiver operating characteristics (ROC) curve of the classifiers are shown in Fig. 3.17(a). The results show that the geometry features outperform the wavelet features while the conventional cartoon features perform the poorest. The overall classification accuracy is 83.5% for the geometry feature, 80.3% for the wavelet feature and 71.0% for the cartoon feature (These numbers are different with 99% statistical significance).

To understand the strengths and the weaknesses of each approach on different

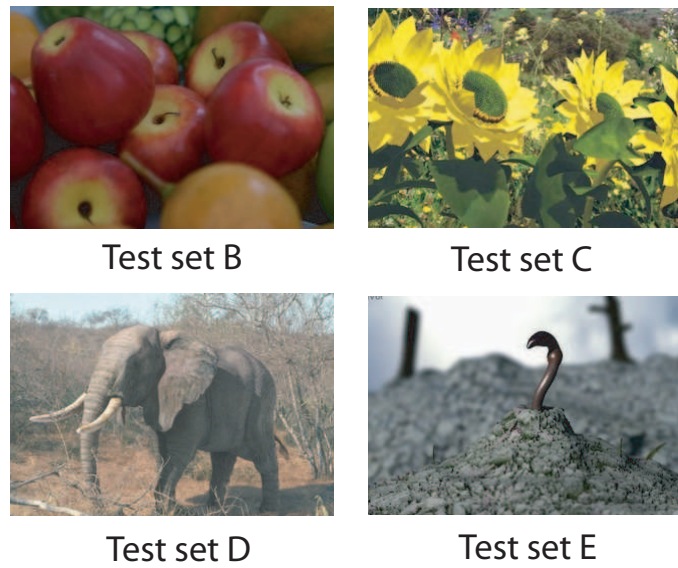


Figure 3.18: Examples of the test image sets in Table 3.1

images, we further test the classifier with images of some interesting and visually confusing categories. Results are shown in Table 3.1. Example images of the test sets are shown in Fig. 3.18. The test accuracy reported is based on the classifier trained with the entire test category held out (i.e., no image of the test category is in the training set). We specifically conduct this experiment in order to study whether a good classifier can be learnt from images of different content categories. Notice that the geometry feature classifier outperforms that of the wavelet feature in three out of the five categories. The poor performance (for both wavelet and geometry features) on test set C indicates that the nature-scene PRCG have unique characteristics which cannot be learnt from the non-nature-scene ones. This could be due to the fact that nature-scene PRCG are mainly generated by the specialized rendering software such as *Terragen* and the nature-scene content often occupies the entire image. In contrast, PRCG with living objects (test set D) have a background which bears the characteristics which can be learnt from other PRCG. The results for the PIM of artificial objects (e.g., wax figures and decorative fruit) of test set B

indicates that the artificiality of the real-world objects does not affect the classifiers. For test set E, the camera DoF effect is a global effect and our classifiers are based on local features, therefore simulating the DoF effect on PRCG does not prevent correct classifications.

In Table 3.1, almost all of the *recaptured CG* (test set A) are classified as PIM, for both sets of feature. Therefore, if we consider recapturing computer graphics as a form of attack to our computer graphic detector, it would be very effective. However, we can form a counter-attack measure by incorporating the *recaptured CG* into the computer graphics category during the classifier learning process. By doing so, the resulting classifiers have a ROC curve as shown in Fig. 3.17(b). Note that the classifier is trained by having a pair of the computer graphics and its recaptured counterpart either entirely in the training set or the test set, it is to prevent the classifier from overfitting to the content similarity of the pairs. Results in Fig. 3.17(b) shows that this strategy renders the recapturing attack ineffective.

The set of geometry features can be decomposed into five sub-groups of features according to different physical motivations. These five sub-groups of features are the surface gradient features (g), the second fundamental form features (s), the beltrami flow features (b), the local patch statistic features (p), and the local fractal dimension features (f). We would like to evaluate the classification performance of each feature sub-group and their combinations. Fig. 3.19 (a) shows the classification accuracy of the feature sub-group combinations excluding the local fractal dimension feature sub-group, where the results are ordered according to their classification accuracy from high to low, whereas Fig. 3.19 (b) shows those with the local fractal dimension feature sub-group included. Note that the local fractal dimension feature sub-group has the lowest classification accuracy at 59.9%, which is consistent with that observed in Fig. 3.14. Furthermore, its role is most insignificant in the complete

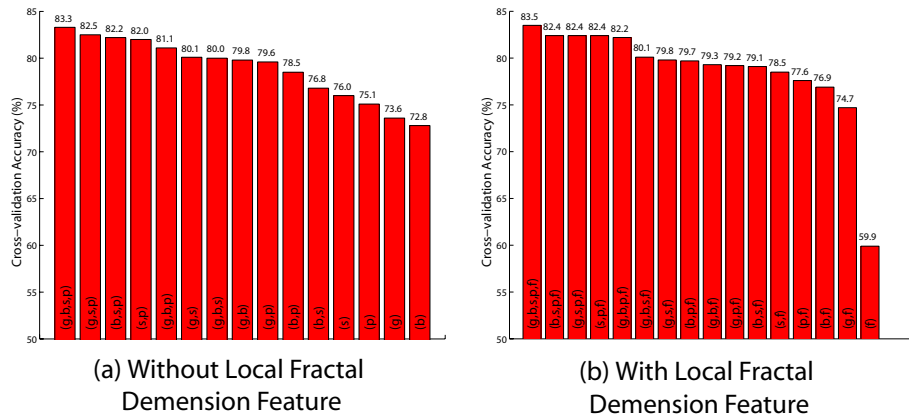


Figure 3.19: Classification performance of feature combinations (a) without and (b) with local fractal dimension features. Legend: g = surface gradient features, b = beltrami flow features, s = second fundamental form features, p = local patch statistics features.

combination of all feature sub-groups, as can be seen from the small classification accuracy difference between 83.5% for the (g,b,s,p,f) combination and 83.3% of the (g,b,s,p) combination, as compared to other feature sub-groups. The weaknesses of the local fractal dimension features are probably due to the fact we do not segment the image into smooth regions and textured regions before computing the statistics of the local fractal dimension. Such pre-segmentation may help improving the features as the fractal dimension of the smooth regions are mainly not interesting. The experimental verification of this suggestion will be considered in the future work. From Fig. 3.19, it seems that good performance (82%) can be achieved even if we use two features only (s and p). g and b does not add much on top of these two.

We also analyze the computational complexity of the features by performing feature extraction on 100 images in Matlab 7.0. Their per-image feature-extraction time in seconds are 9.3s (wavelet), 5.2s (surface gradient), 8.7s (2nd fundamental form), 6.5s (Beltrami flow vector), 3.0s (local patch), 128.1s (fractal). Except for the fractal feature, the other features are quite computationally efficient.

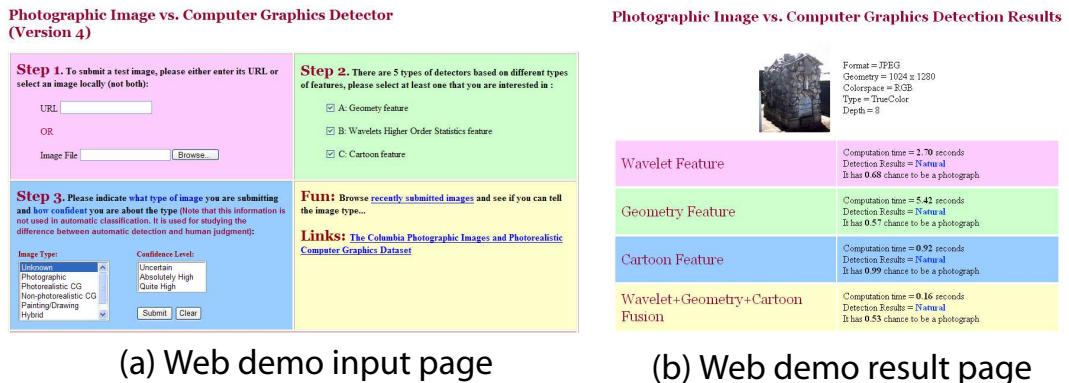


Figure 3.20: The screen capture of the web demo user interface.

### 3.12 An Online Demo System

We deployed an implementation of the geometry features as an online demo system (accessible from [www.ee.columbia.edu/trustfoto](http://www.ee.columbia.edu/trustfoto)) since October 2005 [68]. The goals of the online demo system are to give the web users a more concrete idea on the problem of classifying photographic images and computer graphics, and invite web users to help testing the three classifiers that we deployed, i.e., the geometry classifier (based on the geometry features), the wavelet classifier (based on the wavelet features), and the cartoon classifier (based on the cartoon features). The implementation details for the online classification system is described in Appendix D.

Fig. 3.20 shows the input interface and the result page of the online system. On the input page, web users are required to input the URL of an online image or upload an image from the local computer, provide their judgement or knowledge on the image type together with the judged confidence level, and also select the type of classifier output they wish to see. For the image type, the web users can select unknown, photographic, photorealistic computer graphics, non-photorealistic graphics, painting, hybrid, or others (see Fig. 3.21).



Figure 3.21: The image type for the user labels. The keyword for the image type is given in the bracket.

### 3.13 Discussion

Our approach for PIM and PRCG classification arises from asking a fundamental question of how we should describe images such that PIM and PRCG can be better distinguished. We adopt a geometry-based image description using fractal geometry and differential geometry. Additionally, we sample local patches of the image intensity function to form a patch distribution. The geometry-based approach enables us to uncover distinctive physical characteristics of the PIM and PRCG, such as the CRF of PIM and the sharp structures in PRCG, which has not been possible by using any of the previous techniques. We extract the geometry features using the method of rigid body moments, which can capture the characteristics of a high-dimensional joint distribution. The SVM classifier based on the geometry feature outperforms those in prior work. We also analyze the characteristics of recaptured computer graphics and demonstrate that we can make the recapturing attack ineffective. Furthermore, we identify a subset of PRCG with nature scenes, which remains challenging to classify and demands more focused research.

As a future work, we will consider a more fundamental and detailed modelling

of the 3D scene authenticity using the computer graphics and computer vision techniques. Scene authenticity is an important element for passive-blind image forensics. In our experiment, we have not considered the stratification in computer graphics rendering which can be based on the purpose of the rendering, the types of rendering techniques, and the amount resources used for rendering. For such an experiment, we will need to further collect computer graphics with known sources and with detailed rendering description. This study will not only benefit image forensics, but will also help in evaluating the photorealism for various types of computer graphics, for which the results would be of interest to the computer graphics community.

## Chapter 4

# A Geometric Method for Camera Response Function Estimation using a Single Image

### 4.1 Introduction

In this chapter, we present a geometric method for estimating camera response function (CRF) from a single image. The geometric method can be applied for distinguishing different models of camera, which belongs to an intermediate-level image source identification problem. This geometric method is inspired by the observation described in 3.7.1 that image gradient contains information about CRF.

A camera response function (CRF) maps *image irradiance* (light energy incident on an image sensor) to *image intensity* (output of a camera). In practice, the mapping is a collective effect of various camera internal operations and noise [96]. The capability to estimate the CRF is important, as various photometric computer vision algorithms, such as shape from shading, photometric stereo and so on require scene radiance measurement, which is well represented by image irradiance if the effect of the lens distortion is negligible. If a CRF can be estimated, image intensity can be transformed to image irradiance. Furthermore, a CRF can be thought as a natural watermark for an image, which can be used to assess the authenticity of an



image [38].

CRF's can be estimated from three types of inputs: a set of same-scene images with different but known exposures [14, 58, 64], a single RGB color image [49], or a grayscale image converted from a RGB color image [50]. Estimating the CRF from a single image is an under-constraint problem and an assumption on image irradiance is necessary, e.g., the distribution of the image irradiance value at a local edge region is assumed to be uniform in [50]. Unfortunately, for all previous single-image CRF estimation methods, there is no principled mechanism to identify image regions which are consistent with the assumptions, as verifying the assumptions in the unknown image irradiance is non-trivial.

This work contains three main contributions: we propose a new theoretical-based method for estimating CRF from a single grayscale image or color channel, it is the first work showing experiments on single color-channel images, and we propose a new CRF model.

In this work, we propose a theoretic-based CRF estimation method using *geometry invariants* (GI), which provides a constraint equation to identify the potential *locally planar irradiance points* (LPIP) needed for CRF estimation. The term ‘geometry invariants’ refers to the geometric quantities which are invariant to the geometry variations for a locally planar region on an irradiance image. In the existing literature, the term ‘photometric invariants’ was used to refer to the similar types of geometric quantities on an image [33].

Our method relies on the existence of the locally planar region in image irradiance, which often can be found on the ramp edges in an image. Our method consists of three main computational steps: computing image derivatives, detecting LPIP, and CRF estimation, as shown in Fig. 4.1. In Fig. 4.1, we also show the implementation issues related to the computational steps. Our proposed implementation

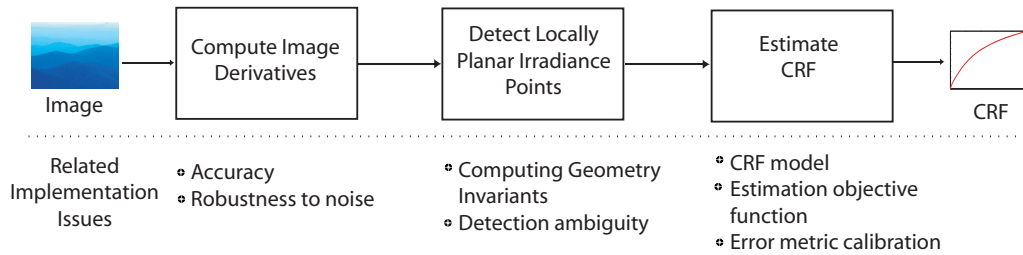


Figure 4.1: The computational steps for our CRF estimation method and their related implementation issues.

is guided by the analysis of GI and hence addresses the issues in a principled way. Our method shows consistent performance over 5 models of camera (from 4 different manufacturers) as compared to three in [49] and two in [50]. Apart from CRF estimation on a single color-channel image, our method can exploit the availability of multiple images from the same camera to provide a better estimation accuracy and stability. In addition, we propose a generalized gamma curve CRF model (GGCM), which empirically fits the real-world CRF’s from DoRF dataset [34] well. Similar to all prior works on CRF modeling, in this paper we assume a spatially uniform CRF over the image, though a spatially-varying CRF may emerge in the future generation of capturing devices.

In this paper, we describe the prior work on CRF estimation in Sec. 4.2. In Sec. 4.3, we present the theoretical aspect of our CRF estimation algorithm, which introduces GI and GGCM. In Sec. 4.6, we describe the implementation aspect of our algorithm, which covers issues such as derivative computation, detection ambiguity, and curve-fitting. Finally, we show our experiments in Sec. 4.7 and conclude with Sec. 4.9. Proofs of the mathematical properties and propositions are included in Appendix E-I.

## 4.2 Prior Works on CRF Estimation

CRF's can be manually estimated by photographing a Macbeth chart with known-reflectance patches, under uniform illumination. CRF's recovered using a Macbeth chart are considered reliable and often accepted as a ground truth for evaluating other CRF estimation techniques [34, 49, 50].

On the other hand, automatic CRF estimation methods relies on assumptions defined in the irradiance domain. As CRF transformation of image irradiance results in a deviation of the assumptions, CRF can be estimated as a function  $f$  which best restores the assumptions. For methods that estimate CRF using multiple same-scene images [14, 58, 64], the exposure ratio among the images provides a relationship between the irradiance images and the CRF is estimated by a function  $f$  that restores this relationship from the intensity image sequence.

In an earlier work [19], a CRF estimation method for a single-channel image is proposed by assuming that non-linear transformation of image irradiance by a *gamma curve*,  $f(r) = r^\gamma$ , introduces correlated harmonics, which can be measured by the magnitude of biocoherence (third-order moment spectra). However, the method is limited to the use of the gamma curve CRF model, which is known to be insufficient for real-world CRF's. In [49], a CRF estimation method using a single RGB-color image by assuming linearly blended edge pixels (between two homogenous regions with distinct irradiance values). When the linear blending assumption holds across the RGB color channels, it can be shown that the image irradiance values at the edge and the adjacent homogenous regions will be colinear in the RGB color space. Additionally, the assumption that the image irradiance values in an image edge region form a uniform distribution was employed to estimate CRF from a grayscale image [50]. However, the method is only demonstrated on grayscale

images converted from RGB color images instead of single-color channel images [50].

### 4.3 Theoretical Aspects of the Algorithm

In this paper, we use  $r(x, y)$  and  $R(x, y)$  respectively for image irradiance and image intensity. CRF is denoted either by  $R = f(r)$ , or  $r = g(R)$ . The 1st-order derivatives  $\frac{\partial R}{\partial x}$  and  $\frac{df(r)}{dr}$  are respectively denoted as  $R_x$  and  $f'(r)$ , with similar convention for higher-order derivatives.

#### 4.3.1 Geometry Invariants

Given  $R(x, y) = f(r(x, y))$ , we take the 1st-order partial derivatives of  $R(x, y)$  and by the chain rule we obtain:

$$DR(x, y) = \begin{pmatrix} R_x & R_y \end{pmatrix} = f'(r) \begin{pmatrix} r_x & r_y \end{pmatrix} \quad (4.1)$$

Note that  $R_x$  is the product of two factors; the first factor  $f'(r)$  is purely related to the CRF while the second factor  $r_x$  is purely related to the geometry of image irradiance. GI can be derived if the second factor, the effect of image geometry, can be removed. Hence, the resulting GI is only dependent on the CRF  $f$  and not the geometry of image irradiance.

It is non-trivial to eliminate the geometry effect of an arbitrary function  $r(x, y)$ . However, a function can be locally approximated by its Taylor expansion, which decomposes the local geometry into polynomials of different orders. The 1st and 2nd-order polynomials are respectively planes and quadratic functions. We can define the 1st-order GI ( $\mathcal{G}_1$ ) as quantities that are invariant to the class of planar

surfaces:

$$\{r(x, y) : r(x, y) = ax + by + c, a, b, c \in \mathbb{R}\} \quad (4.2)$$

For planes, we have  $r_{xx} = r_{xy} = r_{yy} = 0$ , and the 2nd-order partial derivatives of  $R = f(r)$  are given by Eq. 4.3

$$D^2 R(x, y) = \begin{pmatrix} R_{xx} & R_{xy} \\ R_{yx} & R_{yy} \end{pmatrix} = f''(r) \begin{pmatrix} r_x^2 & r_x r_y \\ r_x r_y & r_y^2 \end{pmatrix} \quad (4.3)$$

Then, by taking the ratio of Eq. 4.3 over Eq. 4.1, we obtain  $\mathcal{G}_1$ :

$$\frac{R_{xx}}{R_x^2} = \frac{R_{yy}}{R_y^2} = \frac{R_{xy}}{R_x R_y} = \frac{f''(f^{-1}(R))}{(f'(f^{-1}(R)))^2} = \mathcal{G}_1(R) \quad (4.4)$$

Note that,  $\mathcal{G}_1$ , as a function over  $R$ , depends only on the derivatives of  $f$ , and not the 1st-order geometry of image irradiance (namely, the geometry parameters,  $a$ ,  $b$ , and  $c$  in Eq. 4.2). Such a strict dependence on  $f$  will be explored in this paper to estimate  $f$ . We will refer to the first two equality relations in Eq. 4.4 as *derivative equality constraints* in the rest of the paper, as they imply certain important geometric properties.

### 4.3.2 General Properties of $\mathcal{G}_1$

In this section, we present two general properties of  $\mathcal{G}_1$ , related to the CRF estimation algorithm. Further properties will be described in the later sections.

**Property 4** (Affine Transformation Invariance). *The functional  $\mathcal{G}_1$  is preserved, as the 3-D graph of a planar irradiance  $S = [x, y, r = ax + by + c]^T$  undergoes affine*

transformation:

$$\text{If } AS + B \rightarrow S' \text{ then } \mathcal{G}_1(f(r)) \rightarrow \mathcal{G}_1(f(r')) \quad (4.5)$$

where  $A$  is a  $3 \times 3$  linear transformation matrix, with  $|A| \neq 0$ ,  $B$  is a  $3 \times 1$  translation vector, and  $S' = [x', y', r']^T$ .

Note that, despite the value change from  $r$  to  $r'$ , the underlying function  $\mathcal{G}_1$  remains the same in Eq 4.5. The affine transformation maps a plane to another plane, but as  $\mathcal{G}_1$  is independent of the plane geometry, so it does not change under the transformation. An affine transformation includes rotation, scaling, and translation and is usually imposed on image irradiance when it undergoes white-balancing and contrast adjustment [105]. Therefore, with Property 4,  $\mathcal{G}_1$  is a natural instrument for CRF estimation.

A special case of affine transformation is the rotation of the graph  $S = [x, y, r]^T$  in the  $(x, y)$  plane at a point  $p$  where  $r(p) = r_p$ . In this case, we have  $r_p = r'_p$  and the *value* of  $\mathcal{G}_1$  is preserved:

$$R_{2 \times 2} \begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} x' \\ y' \end{pmatrix} \Rightarrow \mathcal{G}_1(f(r_p)) = \mathcal{G}_1(f(r'_p)) \quad (4.6)$$

where  $R_{2 \times 2}$  is a  $2 \times 2$  rotation matrix. As rotation of image irradiance is equivalent to rotation of the local coordinate frame, then Property 4 also implies that the *value* of  $\mathcal{G}_1$  is preserved under local coordinate frame rotation.

**Property 5** (Integral Solution to CRF). *The partial differential equation (PDE) in*

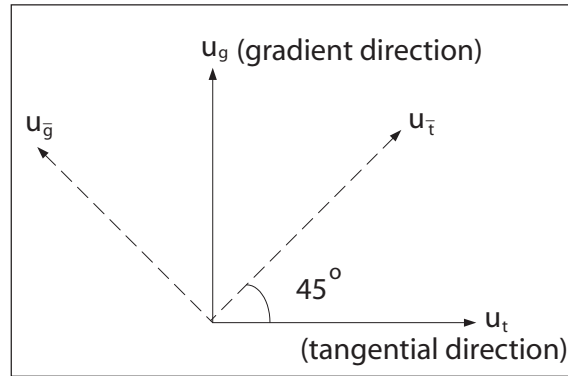


Figure 4.2: The gauge coordinates for computing  $\mathcal{G}_1$ .

Eq. 4.4 can be solved by an integral function of  $\mathcal{G}_1$ :

$$f^{-1}(R) = \int \exp\left(-\int \mathcal{G}_1(R)dR\right) dR \quad (4.7)$$

Despite the above analytical solution for a CRF, its feasibility is in practice deterred by detection ambiguity (Subsec. 4.3.3) and the solution will be approximated by a computational method described later.

### 4.3.3 Detection of Locally Planar Irradiance Points

We have shown that the *derivative equality constraint* of Eq.4.4 is satisfied for every LPIP. Therefore, we may use this constraint to detect candidate points for LPIP in an image. We will also show later that a more general type of surface with linear isophotes also satisfies the equality constraint. We call such inability to uniquely detect LPIP *detection ambiguity*, which will be addressed in Subsec. 4.4.

Property 4 implies in theory that there is not a preferred Cartesian coordinate frame for computing  $\mathcal{G}_1$  because rotation in the local coordinate frame does not change the value of  $\mathcal{G}_1$ . In practice, it is not a good idea to simply compute the  $\mathcal{G}_1$  on the original  $(x, y)$  coordinate frame of an image. The reason is that, for

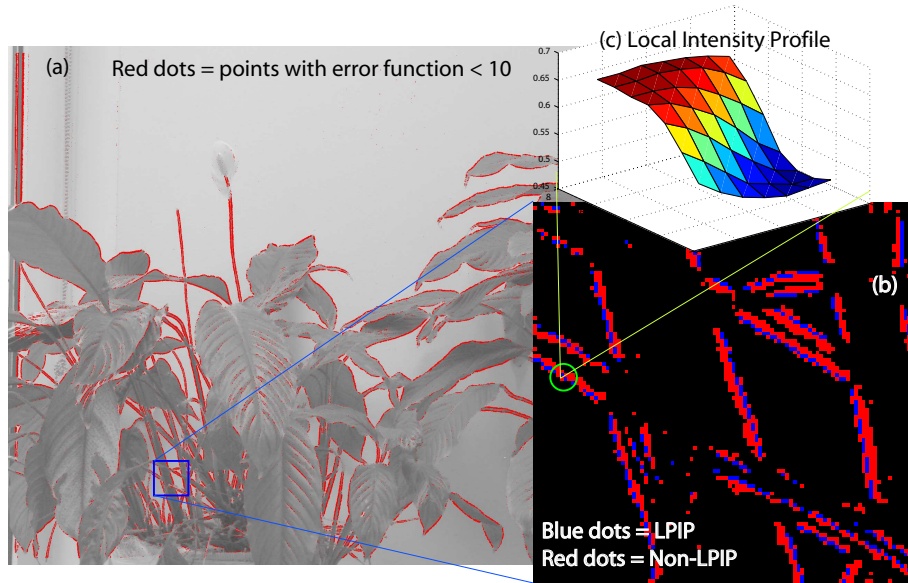


Figure 4.3: (a) Points detected by  $E(R) < 10$ . (b) A magnified view showing the selected points being classifying into LPIP (blue) and non-LPIP (red), where LPIP is often surrounded by non-LPIP. (c) A local intensity profile of an LPIP

example, when an *isophote* (i.e., constant intensity curve) coincides with the  $x$ -axis, a singularity occurs for  $\mathcal{G}_1(R) = \frac{R_{xx}}{R_x^2}$ , as  $R_x = 0$  along an isophote. For computation, we introduce two first-order gauge coordinate frames, which are locally dictated by the intrinsic property of the image function:  $(u_t, u_g)$  and  $(u_{\bar{t}}, u_{\bar{g}})$ -coordinate frames (see Fig. 4.2). The variables  $u_t$  and  $u_g$  represent the local tangential and gradient directions of an image function, where the  $(u_{\bar{t}}, u_{\bar{g}})$ -frame is rotated  $45^\circ$  counter-clockwise from  $(u_t, u_g)$ -frame. Note that the  $(u_{\bar{t}}, u_{\bar{g}})$ -frame stays the farthest possible from the isophote and computing  $\mathcal{G}_1$  on the  $(u_{\bar{t}}, u_{\bar{g}})$ -frame circumvents the above-mentioned singularity problem.

In practice, imposing a strict equality constraint, as in Eq.4.4, is unrealistic in the presence of derivative computation error due to image noise, quantization, and spatial discretization of an image. Therefore, an *error function* in Eq. 4.8 is used to choose the candidate points for LPIP. To simplify notation, we hereforth denote



$R_{u_{\bar{t}}}$  as  $R_{\bar{t}}$ ,  $R_{u_{\bar{g}}}$  as  $R_{\bar{g}}$ .

$$E(R) = \left| \frac{R_{\bar{t}\bar{t}}}{R_{\bar{t}}^2} - \frac{R_{\bar{g}\bar{g}}}{R_{\bar{g}}^2} \right| + \left| \frac{R_{\bar{t}\bar{t}}}{R_{\bar{t}}^2} - \frac{R_{\bar{t}\bar{g}}}{R_{\bar{t}}R_{\bar{g}}} \right| + \left| \frac{R_{\bar{g}\bar{g}}}{R_{\bar{g}}^2} - \frac{R_{\bar{t}\bar{g}}}{R_{\bar{t}}R_{\bar{g}}} \right| \quad (4.8)$$

Our detection method using the above error function is able to detect LPIP points with small spatial support. In computer vision, an image edge profile is often modeled by a one-dimensional ramp, step or peak [99]. It is reasonable to find LPIP on a ramp edge profile, especially at the ramp center, and this hypothesis is empirically validated on an image shown in Fig. 4.3. Fig. 4.3 (a) shows the points detected by  $E(R) < 10$ . Note that, most of the points lie on image edges. In Fig. 4.3 (b), we further classify the detected points (using a method described later) into the LPIP set and the non-LPIP set (formally defined in Subsec. 4.4). Note that, LPIP's are mainly found at the middle of the edges and this supports the above-mentioned hypothesis.

For work in [50], the ramp edge profile in irradiance implies a uniform distribution of edge pixel value, which is used as an assumption for CRF estimation method. In contrast to our method, theirs requires a larger support from the ramp profile for constructing a reliable histogram and lacks a principled technique to detect ramp edges in the irradiance domain.

Finally, to compute the geometry invariant quantities  $\frac{R_{\bar{t}\bar{t}}}{R_{\bar{t}}^2}$ ,  $\frac{R_{\bar{g}\bar{g}}}{R_{\bar{g}}^2}$ , and  $\frac{R_{\bar{t}\bar{g}}}{R_{\bar{t}}R_{\bar{g}}}$  on the  $(u_{\bar{t}}, u_{\bar{g}})$ -coordinate frame, it seems that we have to first find the gradient direction at each local point, and then rotate the local coordinate frames separately according to their gradient direction. Fortunately, there exists a computationally efficient way of computing them. As these geometry invariant quantities are expressed on the gauge coordinates, they become geometric quantities on the image function  $R$  and would have nothing to do with the coordinate frame on which they are computed.

Therefore, these geometry invariant quantities have a general expression for which the computation can be done on any coordinate frame. For instance, on the original  $(x, y)$  coordinate frame, the geometry invariant quantities are given by:

$$\frac{R_{\bar{g}\bar{g}}}{R_{\bar{g}}^2} = \frac{R_x^2(\Delta R - 2R_{xy}) + R_y^2(\Delta R + 2R_{xy}) + 2R_x R_y \bar{\Delta} R}{(R_x^2 + R_y^2)^2} \quad (4.9)$$

$$\frac{R_{\bar{t}\bar{t}}}{R_{\bar{t}}^2} = \frac{R_x^2(\Delta R + 2R_{xy}) + R_y^2(\Delta R - 2R_{xy}) - 2R_x R_y \bar{\Delta} R}{(R_x^2 + R_y^2)^2} \quad (4.10)$$

$$\frac{R_{\bar{g}\bar{t}}}{R_{\bar{g}} R_{\bar{t}}} = \frac{(R_x^2 + R_y^2) \bar{\Delta} R + 4R_x R_y R_{xy}}{(R_x^2 + R_y^2)^2} \quad (4.11)$$

where  $\Delta R = R_{xx} + R_{yy}$  and  $\bar{\Delta} R = R_{xx} - R_{yy}$ .

Note that with the above expressions, the geometry invariant quantities can be efficiently computed.

#### 4.3.4 Geometric Significance of Equality Constraint

The derivative equality constraint specified in Eq.4.4 has an intuitive geometric interpretation. We first introduce three geometric quantities [23] called the isophote curvature ( $\kappa$ ), the normalized 2nd-derivative in the gradient direction ( $\lambda$ ), and the flow line curvature ( $\mu$ ), all expressed in the  $(u_t, u_g)$ -frame, as shown in Eq. 4.12.

$$\kappa = -\frac{R_{tt}}{R_g}, \quad \lambda = \frac{R_{gg}}{R_g} \quad \text{and} \quad \mu = -\frac{R_{tg}}{R_g} \quad (4.12)$$

The basic interpretation of a curve's curvature is the local curve deviation from a tangent line. The isophote curvature  $\kappa$  is the local curve deviation in the isophote direction, where  $\kappa = 0$  indicates a linear isophote. The flow line curvature  $\mu$  is the

local change in the gradient vector field along the isophote, where  $\mu = 0$  indicates same-shape isophotes along the gradient direction (see Fig. 4.4 (b)). Then,  $\kappa = \mu = 0$  if and only if a local region is composed of linear isophotes (as exemplified in Fig. 4).

**Proposition 3** (Decomposition of  $\mathcal{G}_1$ ). *The  $\mathcal{G}_1$  can be decomposed as below:*

$$\frac{R_{\bar{t}\bar{t}}}{R_{\bar{t}}^2} = \frac{\lambda - \kappa - 2\mu}{R_g}, \frac{R_{\bar{g}\bar{g}}}{R_{\bar{g}}^2} = \frac{\lambda - \kappa + 2\mu}{R_g} \text{ and } \frac{R_{\bar{t}\bar{g}}}{R_{\bar{t}}R_{\bar{g}}} = \frac{\lambda + \kappa}{R_g} \quad (4.13)$$

Proposition 3 associates  $\mathcal{G}_1$  with three geometrically meaningful quantities. The decomposition immediately leads to Proposition 4 which puts an equivalence relationship between the equality constraint and the vanishing of the isophote and flowline curvature. This indicates a local geometric structure of a one-parameter function with a linear isophote, which resembles the local image intensity profile shown in Fig. 4.3 (c).

**Proposition 4** (Geo. Significance of Equality Constraint). *The equality constraint in Eq.4.4 implies the vanishing of the isophote curvature  $\kappa$  and the flow line curvature  $\mu$ :*

$$\left\{ \frac{R_{\bar{t}\bar{t}}}{R_{\bar{t}}^2} = \frac{R_{\bar{g}\bar{g}}}{R_{\bar{g}}^2} = \frac{R_{\bar{t}\bar{g}}}{R_{\bar{t}}R_{\bar{g}}} \right\} \iff \{\kappa = \mu = 0\} \quad (4.14)$$

and locally,  $R = R(u_g)$ , i.e., an arbitrary function depends only on  $u_g$  in the gradient axis, with the linear isophote coincides with the tangent axis.

Proposition 4 indicates detection ambiguity by showing that the constraint equation detects points in a general region with linear isophotes, for which the LPIP set is only a subset. In other words, the constraint detects points on both regions of  $f(ax+by+c)$  and  $f(h(ax+by+c))$ , where  $h$  is an arbitrary function. This fact is illustrated in Fig. 4.4. Both  $r(x, y) = ax + by + c$  and  $r(x, y) = h(ax + by + c)$  have linear

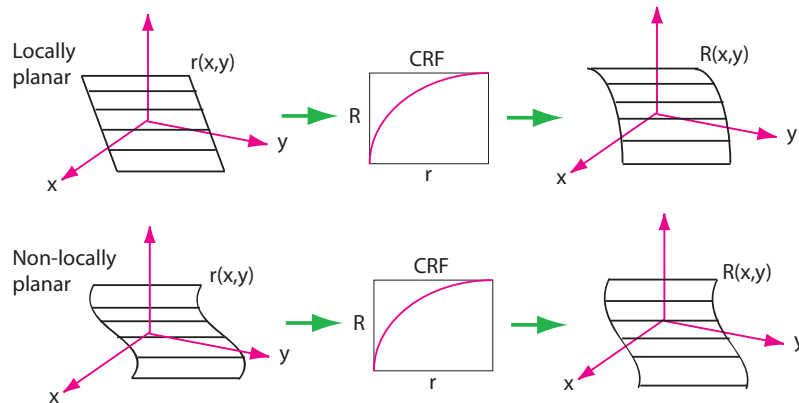


Figure 4.4: The CRF transformation preserves the shape of an isophote. A LPIP point with linear isophotes (as shown in the top row) in the irradiance domain retains the linear isophotes in the intensity domain and hence satisfies the equality constraint. It is possible that there exists a non-LPIP with a linear isophote (as shown in the bottom row) that satisfies the equality constraint. This results in a detection ambiguity in the point selection criterion using the equality constraint.

isophotes. Since the CRF transformation preserves the shape of an isophote, both the corresponding  $R(x, y) = f(ax + by + c)$  (a LPIP) and  $R(x, y) = f(h(ax + by + c))$  (a non-LPIP) have linear isophotes and satisfy the equality constraint. This results in a detection ambiguity in the point selection criterion using the equality constraint. Though non-LPIP points satisfy the derivative equality constraint, they do not satisfy the assumption for  $\mathcal{G}_1$ . In order to improve the LPIP selection, we propose a model-based inferencing method (Subsec. 4.4) to further detect LPIP from the candidate set found by the error function (Eq. 4.8).

### 4.3.5 CRF Estimation Model

Detection ambiguity motivates model-based CRF estimation method; a restricted function space imposed by a CRF model makes CRF estimation more reliable in the presence of noise. We describe the potential CRF models in this subsection. Details of the estimation criteria and procedure presented in Sec. 4.5. Property 6

lead us to a suitable CRF model for our method.

**Property 6** (Relationship with Gamma Curves). *For the CRF  $f$ , from a family of gamma curves,  $R = f(r) = r^\gamma$ ,  $\mathcal{G}_1$  has a simple relationship with the parameter  $\gamma$ :*

$$\mathcal{G}_1(R) = \left( \frac{\gamma - 1}{\gamma R} \right) \text{ and } \gamma = \frac{1}{1 - \mathcal{G}_1(R)R} \doteq Q(R) \quad (4.15)$$

From Property 6, if we adopt gamma curves  $f$  as CRF models, it is not advisable to search for the best  $f$  (and equivalently the best  $\gamma$ ) by fitting the expression for  $\mathcal{G}_1(R)$  in Eq. 4.15, because the expression has a singularity at  $R = 0$ , which will dominate the curve-fitting cost function. Fortunately, Eq. 4.15 suggests that when  $\mathcal{G}_1(R)$  is transformed to  $Q(R)$ ,  $Q(R)$  is equal to  $\gamma$ , a constant function independent of  $R$ , which is also bounded for convex gamma curves,  $\gamma \in [0, 1]$ . Therefore, Property 6 gives us a compatible pair of CRF models and the expression for estimating the CRF.

#### 4.3.5.1 Generalized Gamma Curve Model

Gamma curves are limited in representing real-world CRF's. Therefore, we propose a *generalized gamma curve model* (GGCM) that has a good fit to real-world CRF's (verified on DoRF database [34] with 201 real-world CRF's) for curve-fitting. GGCM provides two representations of CRF with  $f : r \mapsto R$  (called GGCM  $f$ ) and  $g : R \mapsto r$  (called GGCM  $g$ ), as shown in Eq. 4.16.

$$f(r) = r^{P(r, \tilde{\alpha})} \text{ and } g(R) = R^{1/P(R, \tilde{\alpha})} \quad (4.16)$$

where  $\tilde{\alpha} = [\alpha_1, \dots, \alpha_n]$ ,  $P(x, \tilde{\alpha}) = \sum_{i=0}^n \alpha_i x^i$  is a  $n$ -th order polynomial, with  $n + 1$  parameters. The GGCM  $f$  and the GGCM  $g$  model do not form an inverse pair, but both models can be used to represent a camera curve. Note that, GGCM is reduced

Table 4.1: Mean RMSE ( $\times 10^{-2}$ ) of the proposed CRF model

Model	Number of model parameters			
	1	2	3	4
GGCM $f$	5.18	2.34	1.16	0.60
GGCM $g$	8.17	1.46	0.97	0.49
EMOR [34]	4.00	1.73	0.63	0.25
polynomial [64]	7.37	3.29	1.71	1.06

to the gamma curve model when the polynomial is reduced to a constant term. Additionally, the CRF is commonly represented by a function  $f$  with  $f(0) = 0$  and  $f(1) = 1$ . It is reasonable to normalize  $r$  to  $[0, 1]$  because  $r$  can only be recovered with precision up to a linear scaling and an offset of the actual image irradiance.

Various CRF models has been proposed for CRF estimation. One of the earliest models,  $f(r) = \alpha + \beta r^\gamma$ , is borrowed from the photographic emulsion response function [58], essentially a gamma curve after normalization. A general polynomial CRF model is then proposed in [64]. Recently, an empirical EMOR model [34] is obtained from performing principle component analysis (PCA) on 201 real-world CRF's. As the empirical model lacks the differentiable property of an analytic model, it is not suitable for our method. We evaluate GGCM by performing a least squares fit of the model to the 201 real-world CRF's in the DoRF database. The goodness of fit for each CRF is measured by RMSE and is shown in Table 4.1<sup>1</sup>. Note that GGCM performs slightly worse than the empirical EMOR [34] model but outperforms the polynomial CRF model [64], which is a commonly used analytic CRF model.

---

<sup>1</sup>The mean RMSE for the EMOR and the polynomial CRF model are extracted from [34].

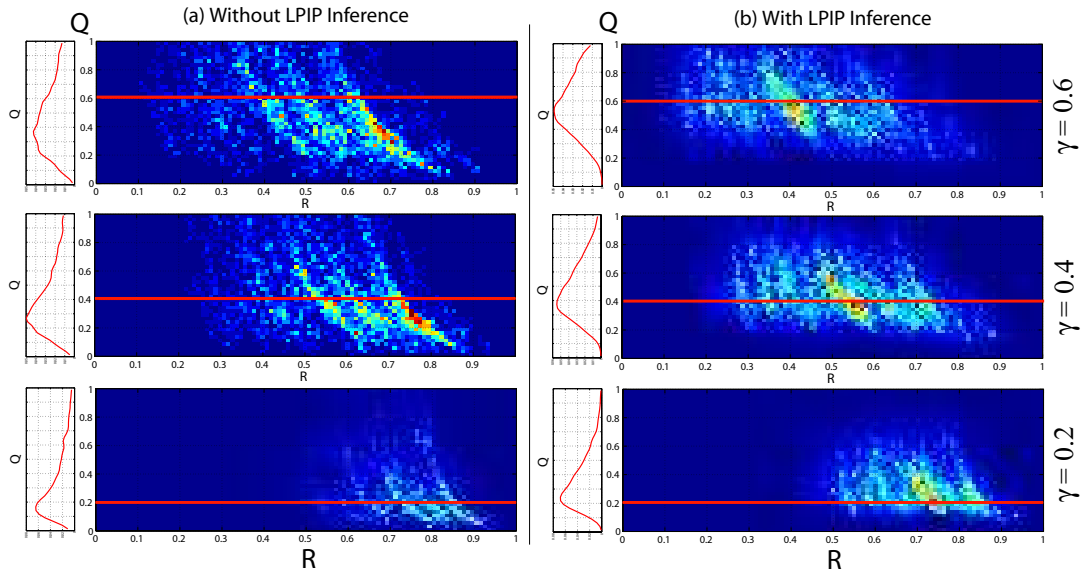


Figure 4.5: The typical  $Q$ - $R$  histogram of LISO from single gamma-curve simulation images with  $\gamma = 0.2, 0.4$  and  $0.6$  for (a) without LPIP inference and (b) with LPIP inference. The red curve on the left is the marginal  $Q$  distribution. The red line in each graph indicates the ground truth value of  $\gamma$

#### 4.4 Addressing Detection Ambiguity Issues

Due to detection ambiguity as mentioned in Subsec. 4.3.4, the equality constraint detects *locally linear isophote points* (LISO), for which LPIP is a subset. As our algorithm requires LPIP for CRF estimation, in this section we present a data-driven approach to infer how likely an LISO to be an LPIP, by exploring the common characteristics of LPIP in terms of their derivative quantities and their spatial layout shown in Fig. 4.3.

To study the effect of detection ambiguity, we generate a set of simulation images by transforming a set of more than 10 irradiance images (extracted from RAW format images which are the direct output from an image sensor) with gamma curves  $f(r) = r^\gamma$  for  $\gamma = 0.2, 0.4$  and  $0.6$ . We observe that more than 92% of the real-world digital camera CRF's in DoRF database lie in between  $r^{0.2}$  and  $r^{0.6}$ . We

define the local isophote (LISO) regions set  $\mathcal{S}_{LISO}$  as:

$$\mathcal{S}_{LISO} = \{(x, y) : E(R(x, y)) < \epsilon\} \quad (4.17)$$

where  $E(R)$  was defined in Eq. 4.8 and  $\epsilon$  is a threshold that can be empirically determined. A soft mapping may also be used in the above to define  $\mathcal{S}_{LISO}$  probabilistically without using a fixed threshold. Here we choose a simple definition and focus on the Bayesian framework of point selection.

Fig. 4.5 (a) shows a typical distribution of the detected LISO points in  $Q$ - $R$  space. Note that from Property 6, the  $Q$  function corresponding to gamma curves is a constant. From the marginal  $Q$  distribution on the left, we see that as  $\gamma$  increases from 0.2 to 0.6, the  $Q$  distribution density consistently shifts to a higher value, as predicted by the theory. However, the mode of the distribution does not coincide exactly with the ground truth value of  $\gamma$  and this is an effect of detection ambiguity, which we will rectify through a LPIP inference.

The goal of the LPIP inference is to identify the LPIP from the non-LPIP. For the LPIP inference, we will perform an independent-feature Bayesian learning using the above-mentioned set of simulated images. In order to ensure the generalizability of the Bayesian learning from the simulated set to the real-world images, we ensure that the simulated images have diverse content (i.e., diverse edge profiles) and multiple gamma curves with  $\gamma$  being 0.2, 0.4, and 0.6 so that the learned statistics are not biased towards the specifics of the data and CRFs used in the simulated pool.

For the LPIP inference, we define two groups of features. The first group consists geometric quantities related to the selection of LISO and the computation of the geometric invariant related quantity  $Q$  (defined in Eq. 4.15). The geometric quantities are the  $E(R)$  value (as defined in Eq. 4.8), the gradient value, and the value



of the normalized 2nd-derivative in the gradient direction ( $\lambda$  defined in Eq. 4.12). These features capture the geometric difference between LPIP and non-LPIP. The second group consists of the moment features that capture the specific spatial layout of LPIP points in the binary LISO map  $b(x, y)$ , where  $b(x, y) = 1$  if  $(x, y) \in \mathcal{S}_{LISO}$ , and  $b(x, y) = 0$  otherwise. Specifically, we compute the 1st to 3rd moment quantities on  $b(x, y)$ , i.e., the total mass  $m_0 = \sum_{W_{5 \times 5}} b(x, y)$ , the centroid  $m_1$ , and the radius of gyration  $m_2$  (Eq. 4.19), in  $5 \times 5$  local windows  $W_{5 \times 5}$ .

$$\begin{pmatrix} m_1^x & m_2^x \\ m_1^y & m_2^y \end{pmatrix} = \frac{1}{m_0} \sum_{(x,y) \in W_{5 \times 5}} \begin{pmatrix} x & x^2 \\ y & y^2 \end{pmatrix} b(x, y) \quad (4.18)$$

$$m_1 = \sqrt{(m_1^x)^2 + (m_1^y)^2} \text{ and } m_2 = \sqrt{m_2^x + m_2^y} \quad (4.19)$$

For the class-dependent feature distributions, we define the LPIP set  $\mathcal{S}_{LPIP}$ , and the non-LPIP set  $\mathcal{S}_{non-LPIP}$  on the simulation images as below:

$$\mathcal{S}_{LPIP} = \{(x, y) : |Q(x, y) - \gamma| \leq 0.1, (x, y) \in \mathcal{S}_{LISO}\} \quad (4.20)$$

$$\mathcal{S}_{non-LPIP} = \{(x, y) : |Q(x, y) - \gamma| > 0.1, (x, y) \in \mathcal{S}_{LISO}\} \quad (4.21)$$

The former condition chooses the points that satisfy the derivative equality constraint (thus in  $\mathcal{S}_{LISO}$ ) and their  $Q$  value are close to the ground-truth value of  $\gamma$ . Only LPIP points meet these conditions simultaneously. The latter condition specifies points in the LISO set but having  $Q$  values distant from the ground-truth  $\gamma$ , thus corresponding to the non-LPIP points. Fig. 4.6 shows the class-dependent feature distribution for  $\mathcal{S}_{LPIP}$  and  $\mathcal{S}_{non-LPIP}$ . Note that,  $\mathcal{S}_{LPIP}$  dominates the low  $E(R)$

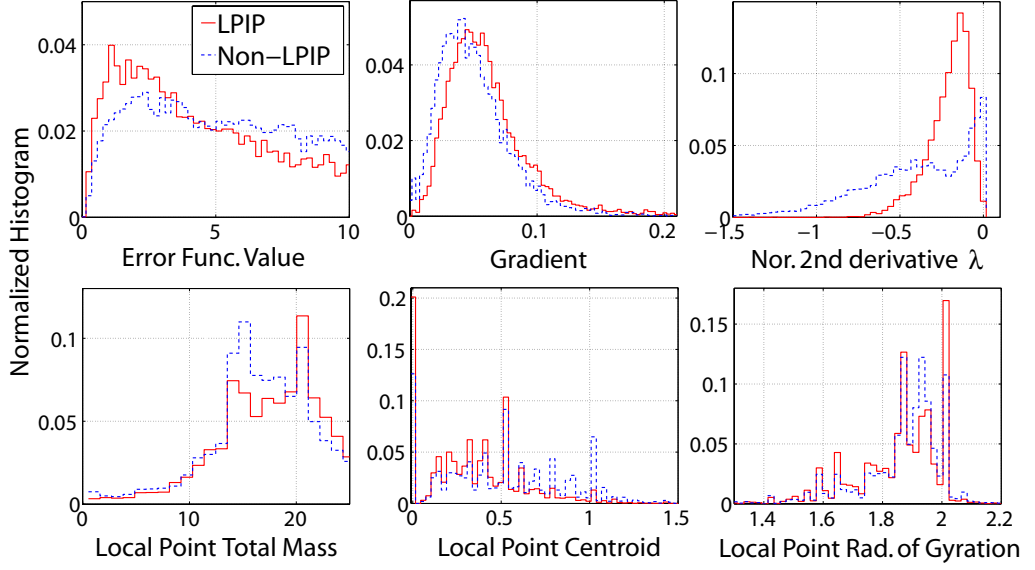


Figure 4.6: The class-dependent feature distributions.

value, as expected by the theory. For the normalized 2nd-derivative feature  $\lambda$ , the high and low values are dominated by  $\mathcal{S}_{non-LPIP}$ , and this is related to the spatial structure of  $\mathcal{S}_{non-LPIP}$  and the higher computational error for these 2nd-derivative values. For the gradient feature,  $\mathcal{S}_{non-LPIP}$  dominates the low value, as regions with low gradient (i.e., flatter) have a lower intensity contrast and therefore tend to suffer more from the quantization noise. On the other hand, the distribution of the moment features derived from the spatial layout of LPIP points can be explained by the spatial structure of  $\mathcal{S}_{LPIP}$  as shown in Fig. 4.3 (b).

For LPIP inference, we adopt a Bayesian approach with enforcement of an independence assumption on features  $f_i$ :

$$P(\tilde{f}|c) = \prod_{i=1}^N P(f_i|c) \text{ where } \tilde{f} = [f_1, \dots, f_N] \quad (4.22)$$

where  $c \in \{\mathcal{S}_{LPIP}, \mathcal{S}_{non-LPIP}\}$ . Feature independence is crucial so that the inference does not capture the specificity of the gamma curves from the geometric features.

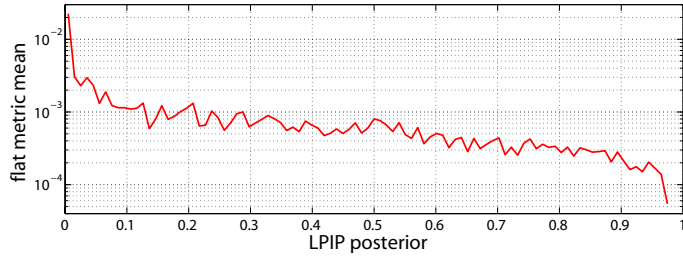


Figure 4.7: Relationship between LPIP posterior and the flatness measurement in an irradiance image. Note the log scale on the y-axis.

For example, the geometry invariants represent a specific relationship between the first and the second derivatives, which are part of the feature set. The a-posterior probability of an LISO being an LPIP is given by:

$$P(c|\tilde{f}) = \frac{P(\tilde{f}|c)P(c)}{P(\tilde{f})} = \frac{P(\tilde{f}|c)P(c)}{\sum_c P(\tilde{f}|c)P(c)} \quad (4.23)$$

where  $P(c)$  is the ratio of points belonging to the class  $c$ .

Fig. 4.5 (b) shows the distribution of LISO in  $Q$ - $R$  space, after incorporating the LPIP posterior as a weight. It is obvious from the marginal distribution  $Q$  that the mode of the distributions coincides very closely to the ground-truth gamma value  $\gamma$ , a sign of overcoming the effect of detection ambiguity. To further validate the effectiveness of LPIP inference point selection, we measure the local flatness of an irradiance image using a simple flatness metric,  $m_f = (r_{xx}^2 + r_{yy}^2)^{0.5}$ , where for a plane  $m_f = 0$ . Fig. 4.7 shows that  $m_f$  decreases (i.e., more flat) as the LPIP posterior increases, and hence verifies the effectiveness of the proposed Bayesian method in selecting the true LPIP points.

## 4.5 CRF Estimation

### 4.5.1 Objective Function for CRF Estimation

Eq. 4.4 expresses  $\mathcal{G}_1$  as a functional of  $f$ . Given  $r = g(R)$  and  $g = f^{-1}$ , we can also express  $\mathcal{G}_1$  in terms of  $g$  as in Eq. 4.24. In this paper, we estimate CRF using  $g(R) = R^{1/(\alpha_0 + \alpha_1 R)}$ , whose  $Q(R)$  is given by Eq. 4.25.

$$\frac{f''(r)}{f'(r)^2} = -\frac{g''(R)}{g'(R)} \quad \text{and} \quad Q(R) = \frac{g'(R)}{1 + g''(R)R} \quad (4.24)$$

$$Q(R) = \frac{(\alpha_0 + \alpha_1 R)^2 (\alpha_1 \ln(R) - \alpha_0 + \alpha_1 R)}{T} \quad (4.25)$$

$$T = \alpha_0^2 + \alpha_0 \alpha_1 R (\alpha_0 (\ln(R) + 1) - 2(1 - \ln(R))) + (\alpha_1 R)^2 (1 - 4\alpha_0 - 2\alpha_1 R + (\ln(R) - 2)(\alpha_1 R + \ln(R))) \quad (4.26)$$

We fit  $Q(R)$  to the computed data  $\{(Q^n, R^n)\}_{n=1}^N$  over  $N$  detected LISO points by minimizing the objective function in Eq. 4.27. Note that the best CRF parameter  $\tilde{\alpha}^*$  is estimated by a weighted least-square criterion, where the weight is the conditional histogram of  $Q$  given  $R$ . The LPIP posterior in Eq. 4.23 are also incorporated. The conditional weight prevents the optimization from being dominated by the data on some specific  $R$ , which happen to be found abundant on an image.

$$\tilde{\alpha}^* = \arg \min_{\tilde{\alpha}} \sum_{j,k} P(Q_j | R_k) |Q_j - Q(R_k, \tilde{\alpha})|^2 \quad (4.27)$$

where  $\tilde{\alpha} = (\alpha_0, \alpha_1)$ ,  $Q_j$  and  $R_k$  are respectively the discrete samples on  $Q$  and  $R$  (representing the histogram bin centers),  $P(Q_j|R_k) = P(Q_j, R_k)/P(R_k)$ , and

$$P(Q_j, R_k) = \sum_{n=1}^N p(\mathcal{S}_{LPIP}|\tilde{f}^n) \mathbf{1}_{[Q_j, R_k]} \{(Q^n, R^n)\} \quad (4.28)$$

where  $[Q_j, R_k]$  is the bin corresponding to the bin center  $(Q_j, R_k)$ , and the indicator function,  $\mathbf{1}_A\{a\} = 1$  if  $a \in A$ . In Eq. 4.28,  $\tilde{f}^n$  are the features extracted from the point associated with  $(Q_j, R_k)$ .

#### 4.5.2 Joint Estimation for Multiple-channel Images

Apart from single-channel images, the proposed method can also be applied to RGB images. Joint estimation of the RGB CRF's can be performed by constraining the similarity between the RGB CRF's, as in Eq. 4.29 with  $\tilde{\alpha} = \{\tilde{\alpha}_r, \tilde{\alpha}_g, \tilde{\alpha}_b\}$ . In practice, the RGB CRF's of a camera are quite similar. Note that, in Eq. 4.29,  $\{R_k\}$  is the set of discrete values for image intensity  $R \in [0, 1]$ .

$$\begin{aligned} \tilde{\alpha}^* = \arg \min_{\tilde{\alpha}} & \left\{ \sum_{j,k,c} P(Q_j|R_k) |Q_j - Q(R_k, \tilde{\alpha}_c)|^2 \right. \\ & \left. + \sum_{c_1 > c_2} \left( \frac{1}{K} \sum_{k=1}^K (g(R_k, \tilde{\alpha}_{c_1}) - g(R_k, \tilde{\alpha}_{c_2}))^2 \right)^{\frac{1}{2}} \right\} \quad (4.29) \end{aligned}$$

Furthermore, if we have  $M$  single-channel images with the same CRF, we can average up their conditional histograms to increase the data coverage on  $R$  (*R-coverage*), as in Eq. 4.30, and then form an objective function as in Eq. 4.27.

$$P(Q_j|R_k) = \frac{1}{M} \sum_{m=1}^M P_m(Q_j|R_k) \quad (4.30)$$

where  $M$  is the number of single-channel images which share a common CRF.

## 4.6 Implementation Aspects of the Algorithm

### 4.6.1 Computation of Image Derivative

There is a large amount of literature covering techniques for computing image derivatives, which includes finite difference, Gaussian scale-space derivatives, and many more. These methods in general work well for common applications like edge detection.

However, our differential method involves computation of derivative ratios from digital images which requires specialized techniques to ensure the computational accuracy and robustness to image noise. There are prior works that involve computing derivative ratios from digital images, such as the works on curve invariants [102] and edge curvatures [99], which use a local polynomial fitting method [59, 99] for computing derivatives that achieves high accuracy in derivative estimation. This method presets the derivative kernel size, and hence has limited adaptability to the wide range of scales in an image. In a similar spirit, we use cubic smoothing B-spline [13] with  $\mathbb{C}^2$  continuity for computing image derivatives in our work. B-spline is a function consists of local piecewise polynomials with a global continuity property. Cubic smoothing B-spline is obtained by simultaneously minimizing a data fitting energy and an  $\mathbb{L}^2$  norm on the second-order partial derivatives (producing smoothing effort).

Compared to local polynomial fitting, we find that cubic smoothing B-spline is more adaptive in terms of image scales and therefore produces considerably more accurate image derivatives. This observation is shown using an experiment conducted on two synthetic images shown in Fig. 4.8, where the increasing-frequency sine func-

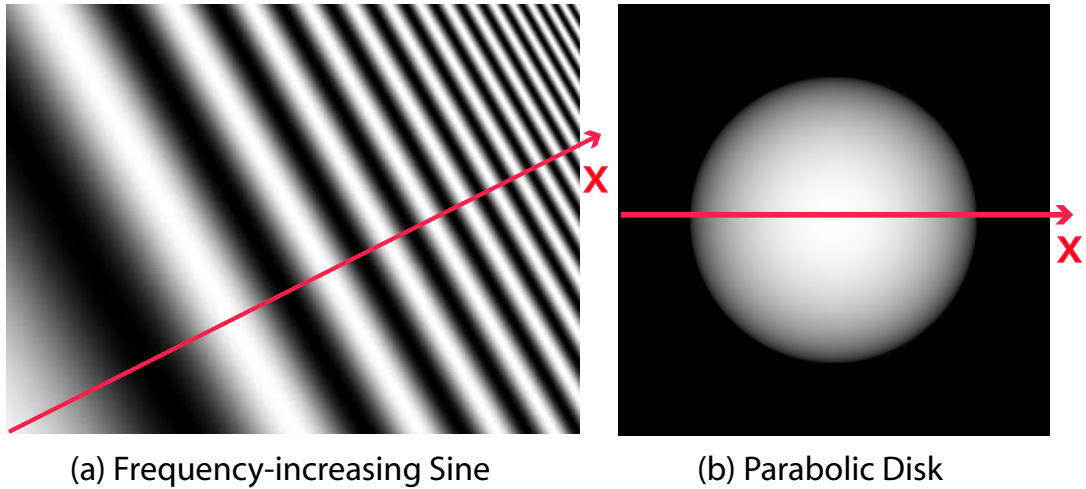


Figure 4.8: (a) The synthetic increasing-frequency sine function image, (b) the synthetic parabolic disk image. The red-color axis  $x$  indicates the line along which the derivative profiles in Fig. 4.9 and Fig. 4.10 are extracted.

tion image (Fig. 4.8 (a)) contains a sine function with a gradually increasing fine image scales, while the parabolic disk image represents a coarse image scale (Fig. 4.8 (b)). Fig. 4.9 and Fig. 4.10 show the derivative computation results respectively for the increasing-frequency sine function image and the parabolic disk image. Note that the local polynomial fitting with a large kernel size works well on coarse image scales, but not on the fine image scales. Whereas the reverse is true for the local polynomial fitting with a small kernel size. However, the smoothing cubic B-spline works well on both scales; in particular, it produces a more accurate estimation for the second-order derivatives.

#### 4.6.2 Error Metric Calibration

Although  $Q(R)$  is compatible with GGCM for CRF estimation through curve-fitting, the space of  $Q(R)$  is not ‘flat’, i.e., its metric is dependent on the CRF curve parameter. For example, RMSE between  $r^{0.1}$  and  $r^{0.2}$  is 0.0465, and that between  $r^{0.5}$  and  $r^{0.6}$  is 0.0771, almost twice of the former, while their  $Q(R)$  RMSE are the

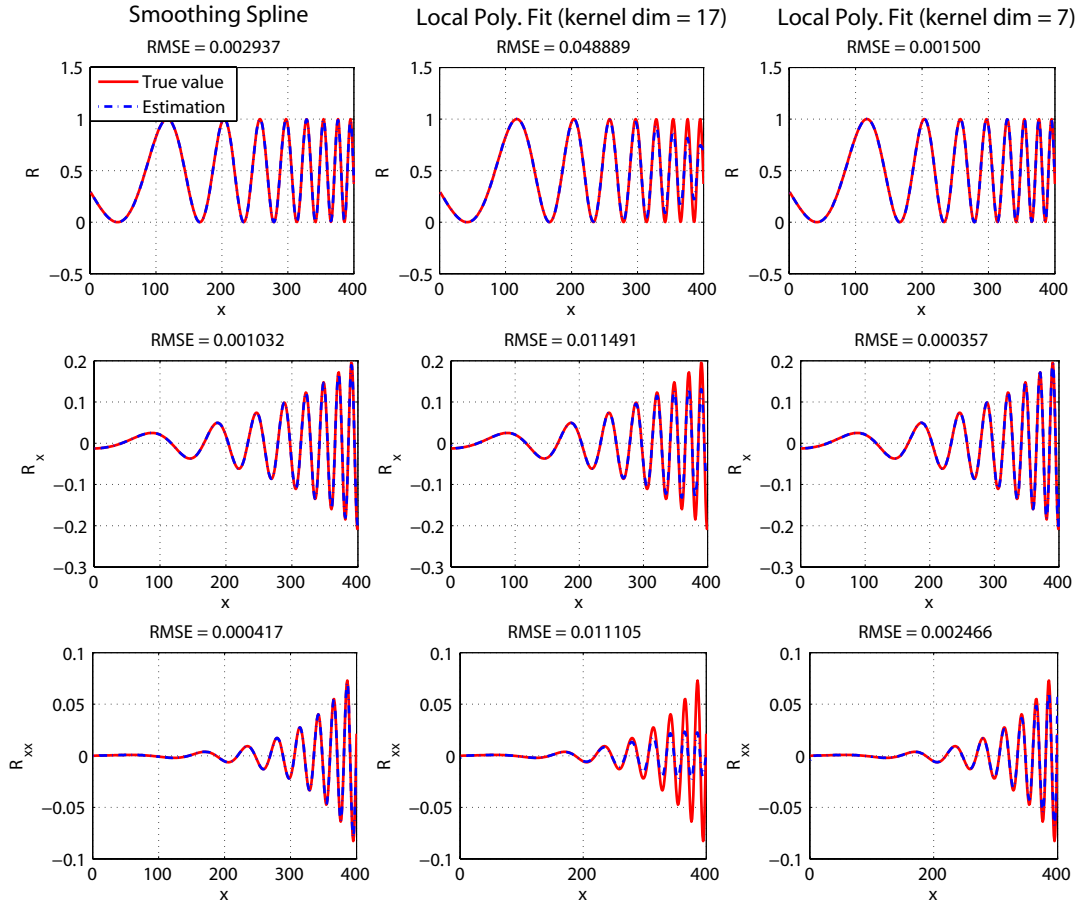


Figure 4.9: The derivation computation results for the synthetic increasing-frequency sine function image in Fig. 4.8 (a). The results shown are the profiles extracted along the red-color axis  $x$  in Fig. 4.8 (a). From the top row to the bottom row are the estimation results of the function  $R$ , its first-order derivative  $R_x$ , and its second-order derivative  $R_{xx}$ . From the left-most column to the right-most column are the estimation results computed by the smoothing cubic B-spline, local 3rd-order polynomial fitting with a  $17 \times 17$  kernel, and the local 3rd-order polynomial fitting with a  $7 \times 7$  kernel.



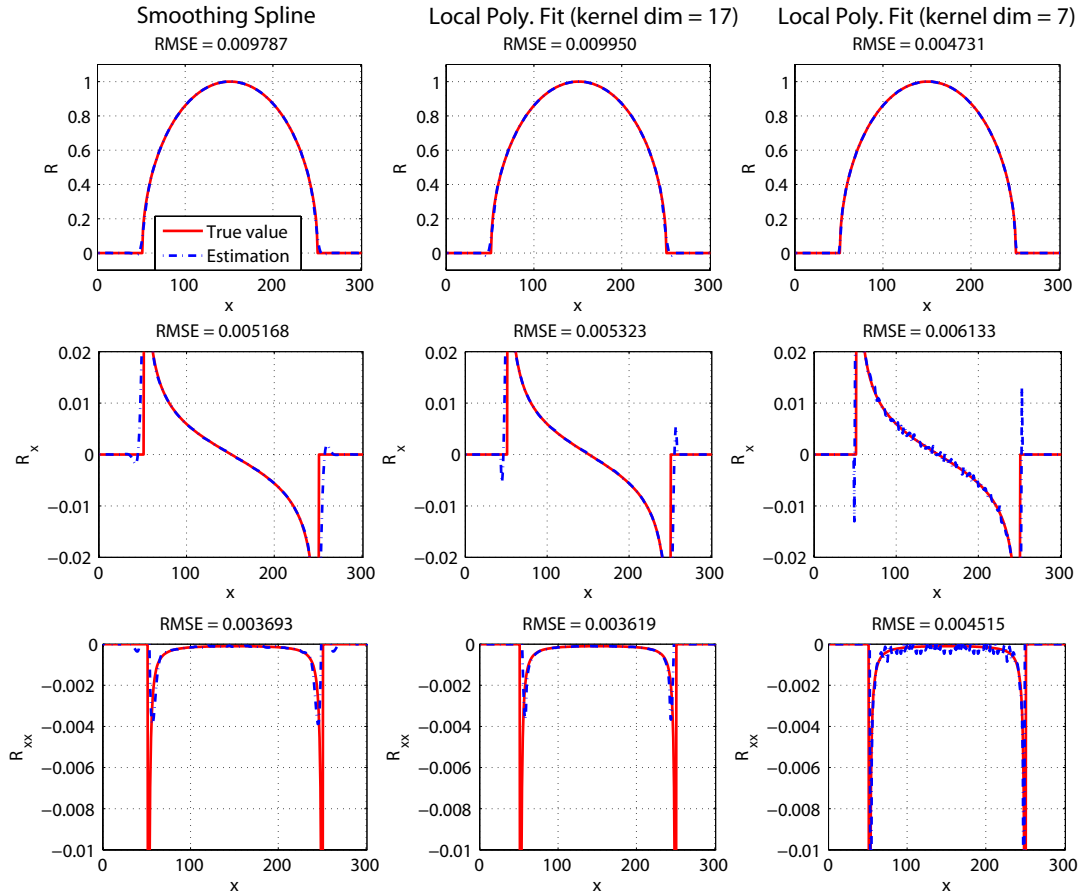


Figure 4.10: The derivation computation results for the synthetic parabolic disk function image in Fig. 4.8 (b). The results shown are the profiles extracted along the red-color axis  $x$  in Fig. 4.8 (b). From the top row to the bottom row are the estimation results of the function  $R$ , its first-order derivative  $R_x$ , and its second-order derivative  $R_{xx}$ . From the left-most column to the right-most column are the estimation results computed by the smoothing cubic B-spline, local 3rd-order polynomial fitting with a  $17 \times 17$  kernel, and the local 3rd-order polynomial fitting with a  $7 \times 7$  kernel.

same. Note that, for gamma curves, RMSE between two  $Q(R)$  is simply  $|\gamma_1 - \gamma_2|$ . In a non-flat space, curve-fitting performance biases towards certain CRF's. Interestingly, the error metric calibration can be formulated as a problem of reparametrizing a space curve with its arc length, which leads to Proposition 5. Such calibration can be done to achieve a linear relationship between the error metric in the  $Q(R)$  space and the error metric in the CRF space (i.e., the RMSE of CRFs). With such a linear relationship, the optimal solution that minimizes the  $Q(R)$  space error is equivalent to the solution that minimizes the CRF error.

**Proposition 5** (Error Metric Calibration). *The error metric in the  $Q(R)$  space can be calibrated with respect to gamma curves,  $f(r) = r^\gamma$ , by a transform on  $Q$ :*

$$\bar{Q} = \frac{\sqrt{3}}{\sqrt{3}-1} \left( 1 - \sqrt{\frac{1}{2Q+1}} \right) \quad (4.31)$$

This calibrated metric can then be used to replace that in Eq. 4.27 to improve the estimation accuracy.

### 4.6.3 Up-weighting Boundary Condition Data

For a differential-based method, data at the center region of  $R \in [0, 1]$  contains information about the center segment of the curve with an unknown additive constant. The end-point data serves as the boundary condition and resolves the additive constant. Therefore, the range of data coverage in  $R$  is important to our CRF estimation method, which is also true for other single-image CRF estimation methods in the prior work [49, 50].

To emphasize the importance of the data corresponding to the boundary condition, we weight the objective function in Eq. 4.27 by a quadratic function  $W$  as

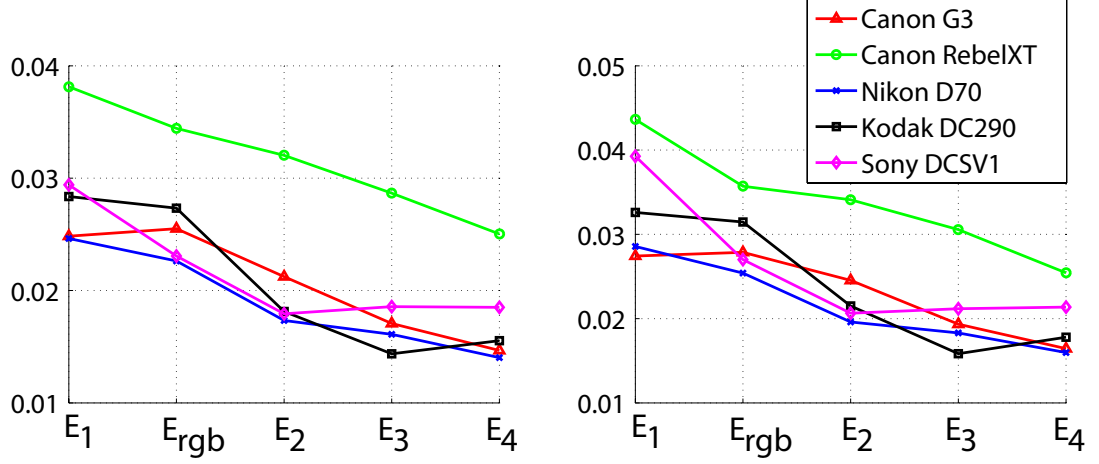


Figure 4.11: A plot of RMSE mean (Left) and RMSE 2nd-moment (Right) for different cameras and different CRF estimation strategies  $E_1$  to  $E_4$  and  $E_{rgb}$ .

Table 4.2: Overall RMSE ( $\times 10^{-2}$ ) for CRF Estimation

Stat.	$E_1$	$E_{rgb}$	$E_2$	$E_3$	$E_4$
Mean	2.91	2.66	2.13	1.90	1.76
2nd Mom	3.43	2.95	2.41	2.10	1.94

below:

$$\tilde{\alpha}^* = \arg \min_{\tilde{\alpha}} \sum_{j,k} W(R_k) P(Q_j | R_k) |Q_j - Q_1(R_k, \tilde{\alpha})|^2 \tag{4.32}$$

where  $W(R_k) = 4(R_k - 0.5)^2 + 1$ . We observe empirically that up-weighting the boundary condition data leads to a more accurate CRF estimation.

### 4.7 Experiments

We test our CRF estimation method using 20 uncompressed RGB-color images (i.e., 60 single-channel images cropped to the size of  $1500 \times 2000$  pixels without filtering or subsampling) from five camera models, i.e., Canon G3, Canon RebelXT, Nikon D70, Kodak DC290, Sony DSCV1, for a total of 100 RGB-images from four

major manufacturers. We select images with at least 80% range coverage in R (*R-coverage*). The importance of R-coverage for CRF estimation is explained in Subsec. 4.6.3.

We estimate the ground-truth CRF for the cameras using a Macbeth chart (using multiple images with different exposures). For each camera, the ground-truth CRF's are indeed similar over RGB channels, with the averaged inter-color-channel CRF difference measured in RMSE being 0.0161. We test our methods using single-color-channel images (denoted as  $E_1$ ), RGB images (denoted as  $E_{rgb}$ ), and also combinations of 2 to 4 single-color-channel images from the same camera (denoted by  $E_2$  to  $E_4$ ). The discrepancy between the estimated and the ground-truth CRF is measured by RMSE. The mean RMSE (measuring accuracy) and the 2nd-moment of RMSE (measuring stability) for the five cameras over RGB color channels and all images of a camera is shown in Fig. 4.11. The overall RMSE mean and RMSE 2nd-moment (over all cameras) are shown in Table 4.2. Note that, both estimation accuracy and stability improve as more images are available, which verifies the importance of R-coverage as combining conditional histograms strictly increases R-coverage.

Fig. 4.11 shows that the estimated CRF's for Canon RebelXT have the least accuracy and stability. As shown in Fig. 4.12 and Fig. 4.13, the estimated CRF's for Canon RebelXT deviate slightly from the knee of the groundtruth curve. Note that, the knee of the Canon RebelXT CRF is very close to linear, and it will be pointed in Sec. 4.8 that our method does not perform well on linear CRF. Fig. 4.12 and Fig. 4.13 respectively show the estimated blue-color channel CRF's for the five models of camera with  $E_1$ , and  $E_4$ . The estimation results for other color channels are similar.

Among all, the CRF of Canon RebelXT and Nikon D70 have the largest differ-

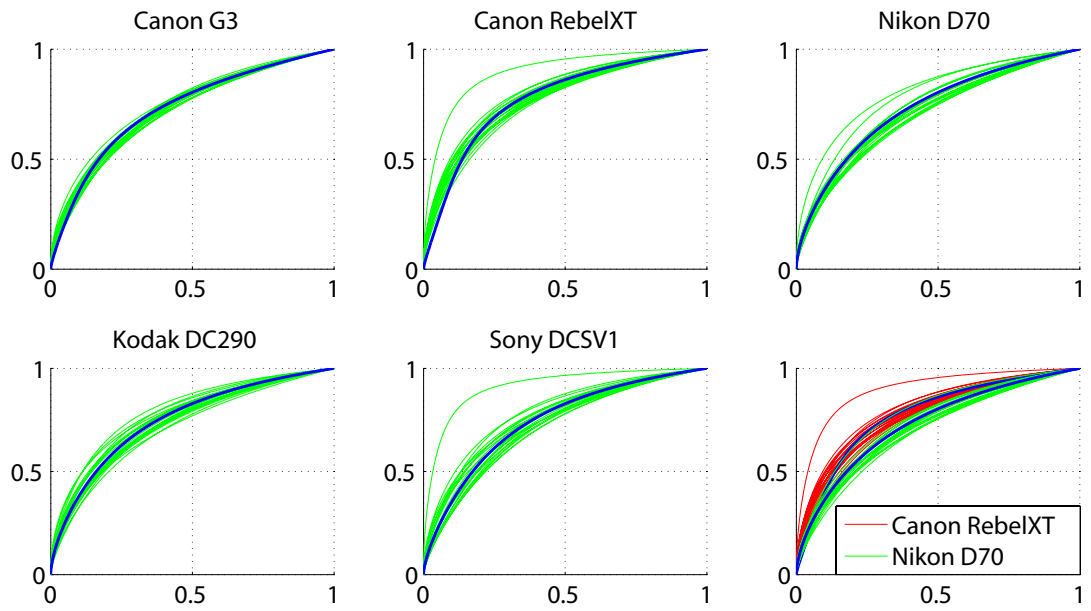


Figure 4.12: Estimated blue-color channel CRF's for the five models of camera using a single color-channel image ( $E_1$ ). The thick blue line represents the ground-truth CRF. The CRF of Canon RebelXT and Nikon D70 are most different and the estimated CRF for Canon RebelXT and Nikon D70 are shown in the lower right subplot.

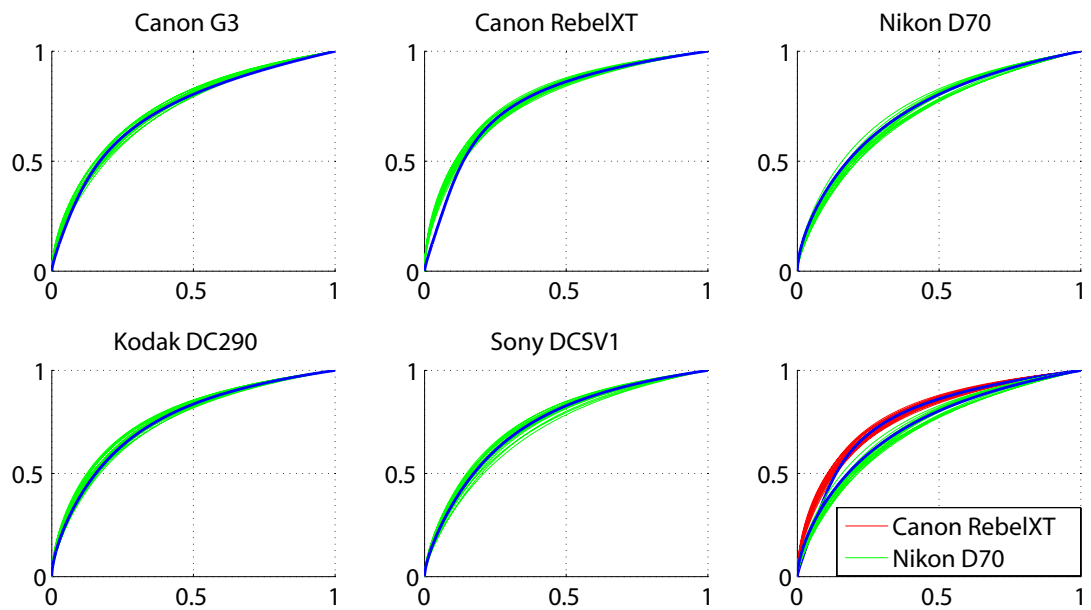


Figure 4.13: Estimated blue-color channel CRF's for the five models of camera using four blue-color-channel images ( $E_4$ ). The thick blue line represents the ground-truth CRF. The CRF of Canon RebelXT and Nikon D70 are most different and the estimated CRF for Canon RebelXT and Nikon D70 are shown in the lower right subplot.

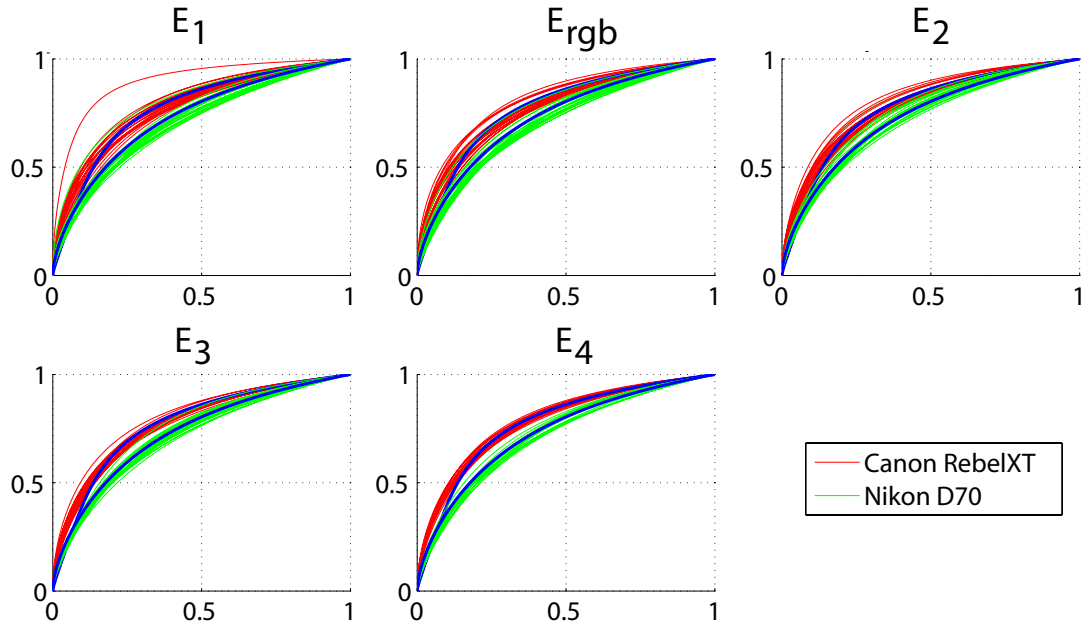


Figure 4.14: Estimated blue-color channel CRF's of Canon RebelXT and Nikon D70 for  $E_1$ ,  $E_{rgb}$ ,  $E_2$ ,  $E_3$  and  $E_4$ . The thick blue line represents the ground-truth CRF.

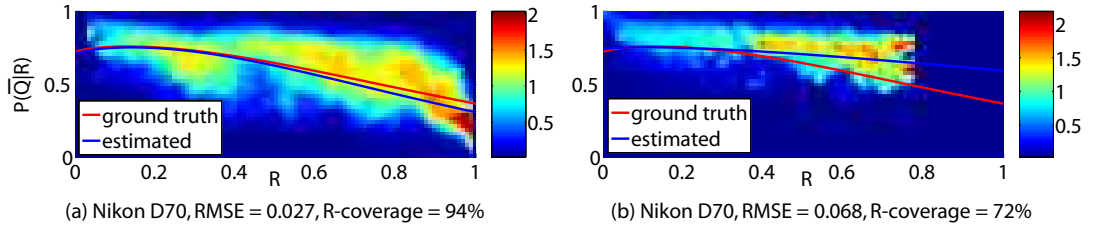


Figure 4.15: Curve-fitting in  $\bar{Q} \times R$  space with data of (a) high R-coverage, (b) Low R-coverage. The thick blue line represents the ground-truth  $Q(R)$  curve.

ence with a RMSE of 0.0781 (averaged over RGB). For a side-by-side comparison, the estimated blue-color channel CRF's for Canon RebelXT and Nikon D70 with  $E_{rgb}$  and  $E_1$  to  $E_4$  are shown in Fig. 4.14. Note that, a slight confusion of the estimated CRF's of the two cameras is observed for  $E_1$ , which is gradually cleared for  $E_{rgb}$ ,  $E_2$ ,  $E_3$ , and  $E_4$ .

Two examples of curve-fitting respectively on data of high and low R-coverage are shown in Fig. 4.15. Note the importance of the end-point data in  $R \in [0, 1]$

which can be seen as the boundary condition for accurate CRF estimation. As experiments in [50] are only conducted on two cameras (different from ours) and the test grayscale images of the digital camera are converted from RGB images (instead of using single-color-channel images), rigorous performance comparison is not possible.

## 4.8 Limitation of the Proposed Method

The proposed CRF estimation method is based on the geometry of the local regions in an intensity image which correspond to the locally planar regions in the corresponding irradiance image. Through CRF transformation, the non-linear shape of the CRF is reflected on the transformed planar regions, and this makes CRF estimation possible through the geometry of the transformed planar regions. Due to this basic principle, the proposed method has two fundamental weaknesses:

1. As the proposed method is based on the transformed planar regions, it is fundamentally incapable of determining whether the CRF of a given image is of a convex shape (e.g.,  $R = r^2$ ) or a concave shape (e.g.,  $R = r^{0.2}$ ), when there exists both locally convex regions and locally concave regions that satisfy the equality constraint (Eq. 4.4) on the image. However, this limitation is not a serious one; for common scene images, an image transformation through a convex CRF would produce an image which is visually different from that transformed by a concave CRF. The statistics of the local geometry of an intensity image can be used to distinguish a convex CRF from a concave one. As most of the CRF for digital cameras are concave, our work begins with an assumption that the CRF shape is concave. A complete CRF estimation algorithm should begin with determining the convexity of the underlying CRF,



which we leave for future work.

2. As a linearly transformed planar region remain planar, the proposed method is incapable of estimating a linear transformation. In practice, the proposed method performs worse when a CRF or a portion of a CRF is close to linear transformation, which is evident in the estimated CRF's for Canon RebelXT as shown in Fig. 4.12 and Fig. 4.13.

Apart from the two fundamental weaknesses, the learning step in the local point selection process results in a preferred range of local geometry on an intensity image, which is used for CRF estimation. We have used gamma transformed images with gamma parameters 0.2, 0.4 and 0.6 as training images. By imposing the feature independence constraint, we attempt to avoid overfitting to the training gamma curves, so that the inference can be generalized to CRF's of more complex shapes, as long as they are within the range of the training gamma curves. However, the inference could not generalize well to CRF's beyond the range of gamma curve. Fig. 4.16 (a) shows the distribution of  $(R, Q)$  points computed from an irradiance image (i.e., a gamma transformed image with  $\gamma = 1$ ), weighted by the LPIP inference learned from gamma images of  $\gamma = 0.2, 0.4,$  and  $0.6$ . Note that, the marginal distribution of  $Q$  peaks at  $Q = 0.7$  but not on  $Q = 1$ . However, we could overcome this problem by enlarging the range of the training CRF by including images with  $\gamma = 1$ . Afterwards, the peak of the marginal  $Q$  distribution shifts to  $Q = 1$ , as shown Fig. 4.16 (b).

In Fig. 4.16 (b), we also see that the peak at  $Q = 1$  is much less prominent, as compared to those shown in Fig. 4.5. This indicates that estimation for CRF's close to linear is increasingly difficult (as pointed out in the second fundamental weakness). In this work, our training set contains images with gamma parameters

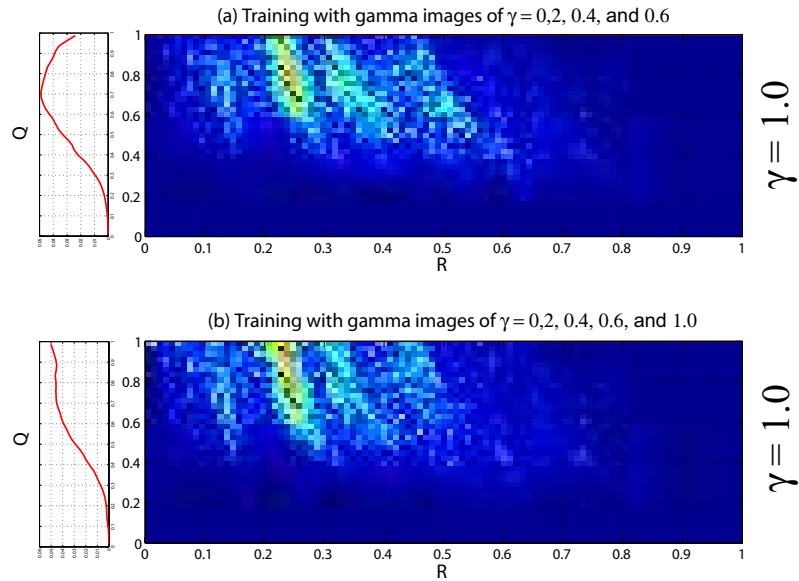


Figure 4.16: The distribution of  $(R, Q)$  points computed from an irradiance image (i.e., a gamma transformed image with  $\gamma = 1$ ). The distribution is represented by a 2D  $R$ - $Q$  histogram obtained from counting the point weight assigned through the LPIP inference. Figure (a) shows the result of the LPIP inference trained using gamma images of  $\gamma = 0.2, 0.4, \text{ and } 0.6$ . Figure (b) shows that from the same training set but with images of  $\gamma = 1.0$  included.

0.2, 0.4 and 0.6 because most of the digital camera CRF's in the DoRF database [34] are within this range.

Furthermore, the technique does not work as well for under/over-exposed images and those with fine texture (examples in Fig. 4.17). The former leads to low  $R$ -coverage and the latter leads to an inaccurate B-spline model used in computing derivatives, as smoothing adversely affects the excessively fine-scale structures.

## 4.9 Discussion

In this paper, we presented a geometry invariant-based method for estimating CRF. In contrast to the single-image CRF estimation methods in prior work, which lack a principled technique to select data consistent to inherent assumptions, our method

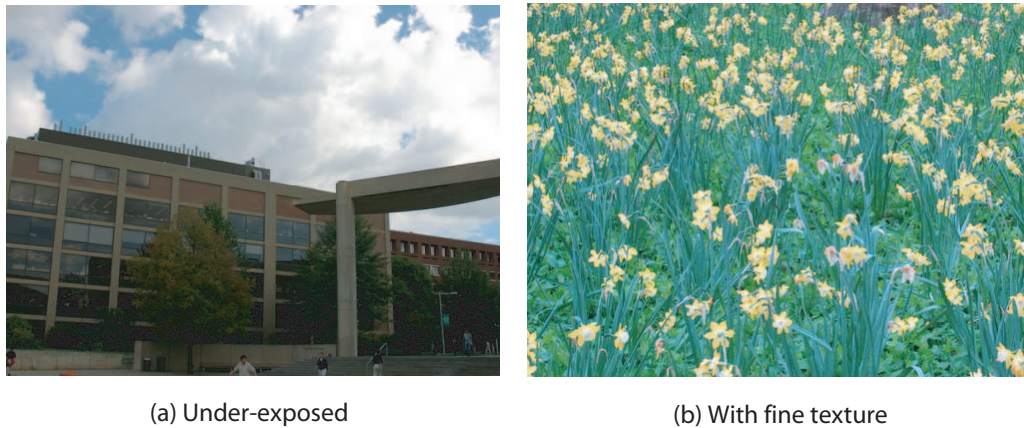


Figure 4.17: Example images that the CRF estimation algorithm perform less well.

provides a theoretical approach and geometrically meaningful criteria for selecting the potential locally planar irradiance points. Comparing to the prior work [50], our results have been shown to be robust over more extensive data and our method is flexible in that we can increase its estimation accuracy and stability when more than one image is available. The geometry invariance theory is novel and may be of wide interest. Techniques in our implementation such as smoothing B-spline for computing image derivatives and the procedure for calibrating the error metric may be useful for other applications. Currently, the algorithm is for the 1st-order geometry invariants, the next step would be to develop an algorithm for the 2nd-order geometry invariants.

## Chapter 5

### Other Related Works

#### 5.1 Active Image Authentication: Digital Signatures and Watermarking

The conventional image authentication techniques belong to the *active approaches* such as digital watermarking and digital signatures [12]. These active techniques rely on prior information such as the pre-added watermark or the pre-extracted content signature.

In the last decade, digital watermarking has been proposed as an active technique for image authentication. The main idea of digital watermarking is to imperceptibly embed a digital watermark onto an image for monitoring image manipulation. Fragile watermarks [26, 103, 104, 106] are sensitive to any minor image modification, while semi-fragile digital watermarks [25, 46, 48, 60] and the content-based digital signatures [7, 9, 47, 61, 85] could accommodate the content preserving operations such as compression and resizing. Unfortunately, the tolerance for the acceptable operations comes at a cost of missing some malicious attacks, apart from the security issue where there is no absolute security for the watermarking secret key. This explains why Friedman's trustworthy digital camera idea [28] for extracting a digital

signature inside a camera, failed to take off.

## 5.2 Passive Image Authentication: Passive-blind Image Forensics (PBIF)

However, for PBIF, no prior information is needed, as indicated by the image forensics process shown in Fig. 2.1. The current research in PBIF can be divided into four main areas of research:

1. Image forgery detection (*Is this image authentic?*).
2. Image source identification (*From what device the image is produced?*)
3. Image operation detection (*What post-processing this image has undergone?*)
4. Counter-attack measure design (*How to handle security attack?*)

Except for the specific works related to ours, which are reviewed in the respective chapters, the other works that are more loosely related to ours are reviewed in this chapter.

### 5.2.1 Image Forgery Detection

There are various works on image forgery detection that are based on the imaging-process authenticity (see Fig. 1.5). For a digital camera, the scene radiance goes through the camera lens before being captured by an array of imaging sensors. The camera lens often has the *optical low-pass* effect for anti-aliasing. The imaging sensors are spatially allocated for measuring three types of colored light. To produce a three-color image, the missing color needs to be interpolated by *demosaiicing*. In order to ensure that a white point in the image scene is rendered as white in the final

image, the color is adjusted through *white-balancing*. The image may go through *enhancements* such as the contrast, sharpness, and saturation adjustment. Finally, *gamma correction* is applied for dynamic range compression and pleasing visual effects.

In [38, 51], the camera response function is estimated to check the consistency between two fragments in an image. Demosaicing results in an interpolation pattern on an image and this pattern can be disrupted when creating image forgery. The work in [81] has used the demosaicing pattern to detect image forgery. On the other hand, the work in [53] uses the estimated fixed pattern noise, a camera signature, to check whether a given image of the same camera has the same noise pattern. A different noise pattern indicates a potential image forgery. However, one can also use the scene authenticity such as scene lighting characteristics for detecting image forgery. The work in [40] estimates point light source direction from the object contours in a single image and examine their consistency for detecting image forgery. The work in [56] shows that lighting consistency can be examined based on spherical harmonics invariant without explicitly estimating the lighting under the assumption of known object geometry. They show an example for checking lighting consistency on a spliced object with two differently illumination parts, without knowing the reflectance property of the object.

### 5.2.2 Image Source Identification

In [21], Farid and Lyu has used natural image statistics (NIS) in the wavelet domain for the forensic verification purpose. They show experiments for distinguishing authentic images from a few other types of images, i.e. stego images (images containing hidden messages), computer graphics, and print-and-scan images. In an ongoing work, we are investigating a number of other NIS, such as NIS in the power

spectrum domain, NIS in the spatial local image patch, and NIS in higher-order statistics, primarily for distinguishing photographic images and computer graphics [65]. To distinguish images captured by different cameras based on their physical device characteristics, one can use camera response function [49, 50] and fixed pattern noise [53]. The former allows one to distinguish different models of camera and the latter different cameras.

### 5.2.3 Image Operation Detection

Image post-processing clues raise suspicion for image forgery and help image forgery detection. In [21], higher-order wavelets statistics are used for detecting image print-and-scan and steganography. Avci et al. has used an image quality measure for identifying brightness adjustment, contrast adjustment and so on [4]. In [93], image operations, such as resampling, JPEG compression, and adding of noise, are modeled as linear operators and estimated by linear image deconvolution. Double JPEG compression has been given special attention in the PBIF literature. In [52, 79], it is observed that double JPEG compression results in a periodic pattern in the JPEG DCT coefficient histogram. Based on this observation, an automatic system that performs image forensics is developed [36]. In [31], it is also found that the distribution of the first digit of the JPEG DCT coefficients can be used to distinguish a singly JPEG compressed image from a doubly compressed one. In [20], the JPEG quantization tables for cameras and image editing software are shown to be different and may serve as a useful forensics clue. As in the case for an image, double MPEG compression artifacts are also observed when a video sequence is modified and MPEG re-encoded [100]. Finally, there are also works on detecting duplicated image fragments due to the copy and paste operation [27, 54, 78].

#### 5.2.4 Counter-attack Measure Design

As it is the goal for a forger to fool the forensic system, a forger can gather information on the forensic system and post-process the forgery so that it escapes the forensic system. Such an action is identified as a *forensic system attack* in Fig. 2.1. Studying the potential forger's attack on a forensic method is necessary before one can design a counter-attack strategy. Apart from the counter-attack measure proposed for the recaptured attack described in Sec. 3.11, the current work is very limited in this aspect. Indeed, there are many possible types of attacks on a forensic system. Below, we will discuss three major types, i.e., oracle attack, recapturing attack, and post-processing attack.

Once a forgery creator has an unlimited access to a forgery detector, an oracle attack can be launched. The forger can incrementally modify the forgery guided by the detection results until it passes the detector with a minimal visual quality loss. In order to make the task of estimating the detection boundary more difficult, the work in [94] proposes a method of converting a parametric decision boundary into a fractal (non-parametric) one, so that an accurate estimation of the boundary requires a much larger number of sample points on the decision boundary. In [97], the oracle attack issue is addressed by modifying the temporal behavior of the detector such that the duration for returning a decision is lengthened when observing a sequence of similar-content input images, which is the hallmark of an oracle attack. The delay strategy can be designed so that the total time needed for an oracle attack to succeed is painfully long.

Apart from the protocol level attack, forgers could apply various post-processing operations to mask image forgery artifacts. This problem can be addressed by the post-processing detection techniques mentioned before. Furthermore, heavy post-



processing is often needed to mask the forgery artifacts.

A more sophisticated post-processing approach would be to simulate the device signature so that the forgery has a consistent device signature. However, such an attack is difficult to implement in practice as the simulated device signature has to be strong enough to mask the inconsistency in the first device signature, and thus results in a tremendous image quality loss.

An attacker can also produce a seemingly authentic image or video by recapturing the sound and sight produced from an image or a video. For example, an image can be printed out and recaptured by a camera. However, such an attack is difficult in practice, as to produce a good quality recaptured duplicate, a subtle and complicated setup for rendering the realistic sound and sight is needed, which is not always feasible. For example, a printed image may contain the perceivable halftoning artifacts and its 2D flatness may lack certain 3D visual effects. Furthermore, recapturing does not remove all the inconsistencies in an image or a video, which is particularly obvious in the scene inconsistencies.

## Chapter 6

### Conclusions

#### 6.1 Summary

This dissertation is dedicated to the research in passive-blind image forensics (PBIF). We identify two main research areas in PBIF as image forgery detection and image source identification, which are directly related to the goal of detecting image manipulation. Other auxiliary research areas are image operation detection, and counter-attack measure design. To approach problems in PBIF, we define two image authenticity properties, i.e., the scene authenticity and the imaging-process authenticity, based on the image formation process starting from the 3D scene at one end to the image acquisition device at the other end. In this dissertation, we present three works for addressing problems in PBIF by capturing the image authenticity properties:

1. We present a statistical method based on bicoherence to capture the optical low-pass property in an image for addressing the image splicing detection problem. We provide a model for image splicing which explains the capability of bicoherence in detecting image splicing. We also propose an additional set of image content-related features to improve the image splicing detection

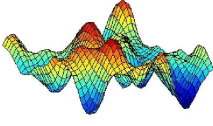
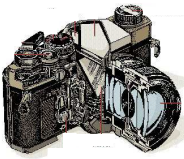
	Techniques	Domain Areas
<p>Statistical Approach</p> 	<p>Authentic signal Modeling            Tampering artifact Modeling            Steganalysis</p>	<p>Statistics, Signal processing,            Machine learning,            Natural image statistics</p>
<p>Physics-based Approach</p> 	<p>Image scene modeling            Image Device modeling            Image Formation modeling            Image manipulation process modeling</p>	<p>Computer vision,            Computer graphics,            Geometry</p>

Figure 6.1: The two main approaches for PBIF: the statistical approach and the physics-based approach.

performance, as compared to that only using the baseline bicoherence features.

2. We present a geometry method based on differential-geometric quantities to capture the properties of object geometry, object surface reflectance, and camera response function (CRF) in images for distinguishing photographic images and photorealistic computer graphics. In contrast to the statistical method in the prior works, the geometry method reveals the physical differences between photographic images and photorealistic computer graphics.
3. We present a geometry method based on geometry invariants to estimate CRF from a single color-channel image. The geometry method offers a novel and principled way for selecting local regions for estimating the CRF from a single greyscale/color-channel image. This method has been shown to be robust over a large, diverse set of test images.

## 6.2 Future Work

PBIF provides an alternative to the active image authentication techniques, such as digital watermarking and digital signatures, for detecting image manipulation. PBIF is a new area of research and poses a lot of challenging research problems. Below are some suggestions for the future work:

1. **Fusion of the Statistical Approach and the Physics-based Approach:**

In general, PBIF can be addressed by two main approaches: the statistical approach and the physics-based approach. Fig. 6.1 shows the techniques and the required domain knowledge for the two approaches. Methods that follow the statistical approach are such as those using natural image statistics [21], modeling tampering artifacts [4, 93] or based on steganalysis-inspired techniques [10, 30], while methods for the physics-based approach are such as those based on the 3D scene properties [40, 56, 70] and the image acquisition device properties [38, 53, 80]. Our works belong to the physics-based approach in the sense that our definition of image authenticity is based on the physical image formation process. The two approaches are complementary and could be combined for inventing new and powerful PBIF methods. Very few current works have considered such a fusion.

2. **3D Scene Consistency:** Despite the few recent prior work having been proposed [40, 56], checking the consistency in the 3D scene remains one of the most difficult areas in PBIF. The difficulties lie in the fact that extracting 3D scene information, such as scene geometry, illumination and surface reflectance property, from a single image is often ill-posed and has no unique solution. However, progress in the area of computer vision have seen novel techniques for 3D scene information extraction being proposed. For instance, the work in [74]

proposed a technique to extract the scene illumination for the image of human eyes. With this technique, we may be able to extract the scene illumination from the eyes of two human subjects in the same image, as illustrated in Fig. 6.2.

3. **Real-world Tampering:** Current works in PBIF seldom demonstrate their capability in detecting image manipulation in real-world cases, where images are often processed over multiple stages with a strategic manner so that the image forgery artifacts become highly imperceivable. This shortcoming is understandable as PBIF research is still at its infancy. In order to bring the innovation in PBIF closer to solving the real-world problems, the future work in PBIF needs to put emphasis in their capability for handling sophisticated image manipulation techniques and multi-stage image manipulation.
4. **Multimodal Forensics:** Current works in passive-blind forensics are mainly limited to the image modality. Works that consider other modalities [18, 100] such as the audio and video modality are very limited. The current image forensics techniques can be extended to other modalities and enable a wider application.
5. **Multi-image Forensics:** Current works in PBIF limit themselves by performing forensic analysis on a single image. Such limitation is indeed artificial. For example, as shown in Fig. 6.3, we can develop techniques to verify the authenticity of the Great Pyramids image shown in Figure 1.1 using other images of the pyramids found using Google Image Search. As a related work, a paper on Photo Tourism [90] has demonstrated the possibility of using multiple unregistered images of the same scene to recover the 3D scene geometry. We may name such PBIF technique using multiple images as *image correlation*.

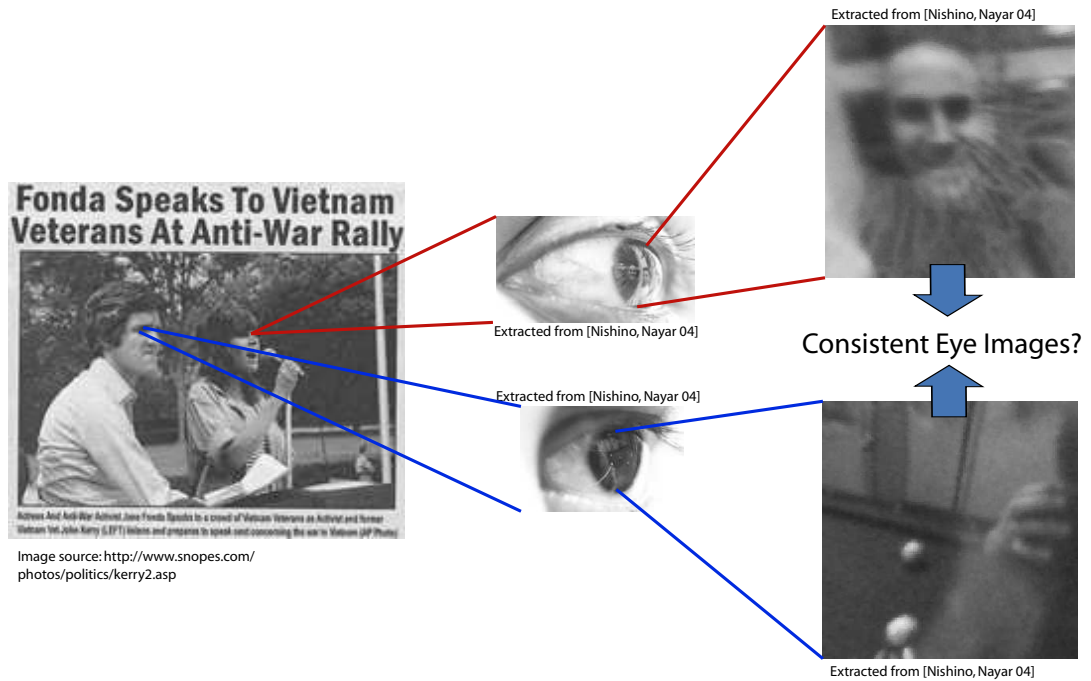


Figure 6.2: An illustrative example of checking the consistency of the scene illumination extracted from the eye image of two different human subjects in the same image.

Image correlation is in fact one of the intuitive ways for human to perform PBIF in such a scenario. Image correlation will enable us to solve a wider range of PBIF problems, offer new solutions to the existing PBIF problems, and offer ways to increase the reliability of the current PBIF solutions.

6. **Camera Evolution:** An issue for the physics-based approach is that the assumed camera model is based on today's technology, which may evolve as technology advances. Future work will need to take such technology evolution into account.

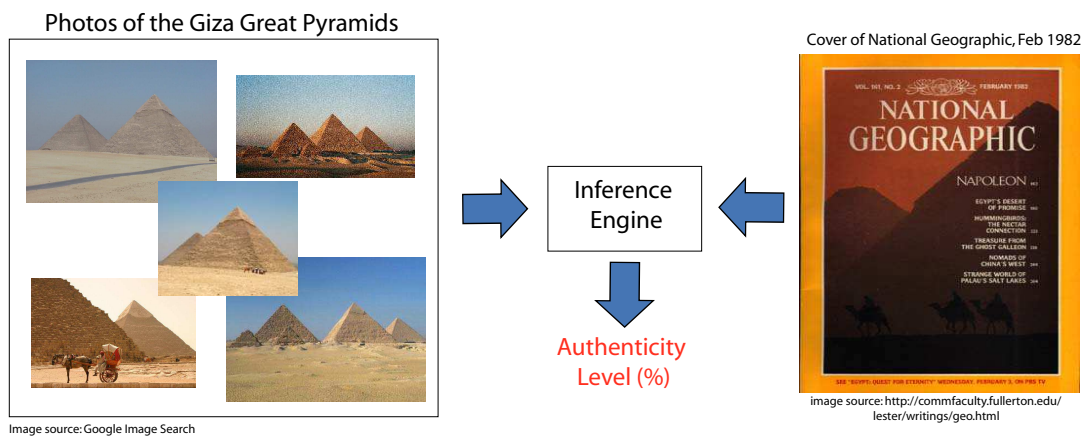


Figure 6.3: An illustrative example of performing PBIF using more than a single image.

## Appendix

### A Proof for the Bipolar Effect on the Phase of the Spliced Signal Bicoherence Proposition (Proposition 1)

The Fourier transform of a spliced signal is given by:

$$S(\omega) = A(\omega) + D(\omega) \quad (\text{A.1})$$

hence

$$S(\omega_1)S(\omega_2)S^*(\omega_1+\omega_2) = A(\omega_1)A(\omega_2)A^*(\omega_1+\omega_2) + C_{ad}(\omega_1, \omega_2) + D(\omega_1)D(\omega_2)D^*(\omega_1+\omega_2) \quad (\text{A.2})$$

where

$$\begin{aligned} C_{ad}(\omega_1, \omega_2) &= A^*(\omega_1 + \omega_2)(A(\omega_1)D(\omega_2) + D(\omega_1)A(\omega_2) + D(\omega_1)D(\omega_2)) \\ &+ D^*(\omega_1 + \omega_2)(A(\omega_1)A(\omega_2) + A(\omega_1)D(\omega_2) + D(\omega_1)A(\omega_2)) \end{aligned} \quad (\text{A.3})$$

As  $C_{ad}(\omega_1, \omega_2)$  consists of cross terms from  $A(\omega)$  and  $D(\omega)$ , we assume that it does not have a systematic effect on  $S(\omega_1)S(\omega_2)S^*(\omega_1 + \omega_2)$ . Then, we can group  $A(\omega_1)A(\omega_2)A^*(\omega_1 + \omega_2) + C_{ad}(\omega_1, \omega_2)$  as  $T_A(\omega_1, \omega_2)$ . If the probability of seeing bipolar signal in an overlapping segment is  $p_d \in [0, 1]$ , then the numerator of the



spliced signal bicoherence  $B_S(\omega_1, \omega_2)$  is given by:

$$\begin{aligned} E[S(\omega_1)S(\omega_2)S^*(\omega_1 + \omega_2)] &= E[T_A(\omega_1, \omega_2)] + E[D(\omega_1)D(\omega_2)D^*(\omega_1 + \omega_2)] \\ &= \bar{T}_A(\omega_1, \omega_2) + p_d 8jk^3 \sin\left(\frac{1}{2}\Delta\omega_1\right) \sin\left(\frac{1}{2}\Delta\omega_1\right) \sin\left(\frac{1}{2}\Delta(\omega_1 + \omega_2)\right) \end{aligned} \quad (\text{A.4})$$

where we represent  $E[T_A(\omega_1, \omega_2)]$  as  $\bar{T}_A(\omega_1, \omega_2)$ . Then, the phase of  $B_S(\omega_1, \omega_2)$  is given by:

$$\phi(B_S(\omega_1, \omega_2)) = \phi(E[S(\omega_1)S(\omega_2)S^*(\omega_1 + \omega_2)]) \quad (\text{A.5})$$

$$= \tan^{-1} \left( \tan(\phi(\bar{T}_A(\omega_1, \omega_2))) + \left( \frac{p_d k^3}{|\bar{T}_A(\omega_1, \omega_2)|} \right) \frac{8T_{\sin}}{\cos(\phi(\bar{T}_A(\omega_1, \omega_2)))} \right) \quad (\text{A.6})$$

where  $T_{\sin} = \sin(\frac{1}{2}\Delta\omega_1) \sin(\frac{1}{2}\Delta\omega_1) \sin(\frac{1}{2}\Delta(\omega_1 + \omega_2))$ . Note that when  $|p_d k^3| \rightarrow 0$ ,  $\phi(B_S(\omega_1, \omega_2)) \rightarrow \phi(B_A(\omega_1, \omega_2))$ , whereas when  $\left| \frac{p_d k^3}{\bar{T}_A(\omega_1, \omega_2)} \right| \rightarrow \infty$ ,  $\phi(B_S(\omega_1, \omega_2)) \rightarrow \pm 90^\circ$  and the effect of bipolar signals increases with  $p_d$ . Therefore, by continuity, the effect of an additive bipolar signal is that it induces  $\pm 90^\circ$  phase concentration on the resulting spliced signal bicoherence and its strength increases with  $|k|$  and  $p_d$ .

## B Proof for the Bipolar Effect on the Magnitude of the Spliced Signal Bicoherence Proposition (Proposition 2)

From definition 3, a spliced signal is modeled as:

$$s(x) = a(x) + d(x) \Leftrightarrow S(\omega) = A(\omega) + D(\omega) \quad (\text{B.7})$$

To simplify analysis, we first assume that every overlapping signal segment has a bipolar signal, i.e., probability of seeing a bipolar signal in a overlapping segment

$p_d$  is 1. Then,

$$\begin{aligned}
& |B_S(\omega_1, \omega_2)| \\
&= \frac{|E[(A(\omega_1)+D(\omega_1))(A(\omega_2)+D(\omega_2))(A^*(\omega_1+\omega_2)+D^*(\omega_1+\omega_2)))]|}{\sqrt{E[|(A(\omega_1)+D(\omega_1))(A(\omega_2)+D(\omega_2))|^2]E[|A^*(\omega_1+\omega_2)+D^*(\omega_1+\omega_2)|^2]}} \\
&\geq \frac{E[||A(\omega_1)|-|D(\omega_1)|| \ ||A(\omega_2)|-|D(\omega_2)|| \ ||A^*(\omega_1+\omega_2)|-|D^*(\omega_1+\omega_2)||]}{\sqrt{E[(|A(\omega_1)|+|D(\omega_1)|)(|A(\omega_2)|+|D(\omega_2)|)]^2}E[(|A^*(\omega_1+\omega_2)|+|D^*(\omega_1+\omega_2)|)^2]} \\
&= \frac{E[|D(\omega_1)||A(\omega_1)-1| \ |D(\omega_2)||A(\omega_2)-1| \ |D^*(\omega_1+\omega_2)||A^*(\omega_1+\omega_2)-1|]}{\sqrt{E[(|D(\omega_1)|(|A(\omega_1)+1)|D(\omega_2)|(|A(\omega_2)+1)|)^2]E[(|D^*(\omega_1+\omega_2)|(|A^*(\omega_1+\omega_2)+1)|)^2]}} \\
&= L(\omega_1, \omega_2) \tag{B.8}
\end{aligned}$$

where  $L(\omega_1, \omega_2)$  is the lower bound for  $|B_S(\omega_1, \omega_2)|$ . Applying Markov inequality, we obtain:

$$P\left(\frac{|A(\omega)|}{|D(\omega)|} \geq \epsilon\right) \leq \frac{E[|A(\omega)|]}{|D(\omega)|\epsilon} \tag{B.9}$$

$$= \frac{E[|A(\omega)|]}{8|k^3 \sin(\frac{1}{2}\Delta\omega_1) \sin(\frac{1}{2}\Delta\omega_1) \sin(\frac{1}{2}\Delta(\omega_1 + \omega_2))|\epsilon} \tag{B.10}$$

For any  $\epsilon > 0$ , as  $|k| \rightarrow \infty$ ,  $\frac{E[|A(\omega)|]}{|k^3|} \rightarrow 0$ , if we assume that  $a(x)$  is an energy signal. Being an energy signal, its energy is finite, and hence,  $E[|A(\omega)|] \leq E[|A(\omega)|^2] \leq \infty$ . Therefore, as  $|k| \rightarrow \infty$ ,  $\frac{|A(\omega)|}{|D(\omega)|} \rightarrow 0$  in probability and hence

$$L(\omega_1, \omega_2) \rightarrow \frac{|D(\omega_1)||D(\omega_2)||D^*(\omega_1 + \omega_2)|}{\sqrt{(|D(\omega_1)||D(\omega_2)|)^2|D^*(\omega_1 + \omega_2)|^2}} = 1 \text{ in probability}$$

As  $L(\omega_1, \omega_2) \leq |B_S(\omega_1, \omega_2)| \leq 1$ ,  $|B_S(\omega_1, \omega_2)|$  also approaches 1 in probability, as  $L(\omega_1, \omega_2)$  approaches 1 in probability when  $|k| \rightarrow \infty$ . The analysis above is the limit case when  $p_d \rightarrow 1$ . By continuity,  $|B_S(\omega_1, \omega_2)|$  increases in probability when  $p_d$  and  $|k|$  increases.

## C Derivation for Gradient on Surface

For any smooth real-valued function  $f$  on a Riemannian manifold  $(M, g)$ , the gradient of  $f$ , denoted as  $\text{grad}f$  at a point  $p$  satisfies:

$$\langle \text{grad}f, V \rangle_g = df(V) \quad (\text{C.11})$$

where  $df$  is a differential of  $f$  and  $V \in T_pM$ . Then,  $\text{grad}f$  can be written as below in smooth coordinates  $(u^1, \dots, u^m)$  with  $\dim(M) = m$ :

$$\text{grad}f = \sum_{ij} g^{ij} \frac{\partial f}{\partial u^i} \frac{\partial}{\partial u^j} \quad (\text{C.12})$$

where  $\frac{\partial}{\partial u^j}$  represents the basis vectors for  $T_pM$ . Hence, the magnitude of  $\text{grad}f$  on  $T_pM$  is given by:

$$|\text{grad}f|^2 = \langle \text{grad}f, \text{grad}f \rangle_g = \sum_{ijpq} g_{ij} g^{pj} \frac{\partial f}{\partial u^p} g^{qi} \frac{\partial f}{\partial u^q} = \sum_{pq} g^{pq} \frac{\partial f}{\partial u^p} \frac{\partial f}{\partial u^q} \quad (\text{C.13})$$

For a graph manifold  $F(x, y) = (x, y, L(x, y))$ , we have:

$$g^{-1} = (g^{ij}) = \frac{1}{1 + L_x^2 + L_y^2} \begin{pmatrix} 1 + L_y^2 & -L_x L_y \\ -L_x L_y & 1 + L_x^2 \end{pmatrix} \quad (\text{C.14})$$

Then, with a real-valued function  $f(x, y) = L(x, y)$ , we obtain:

$$\begin{aligned}
|\text{grad}L|^2 &= g^{xx}(L_x^2) + 2g^{xy}L_xL_y + g^{yy}(L_y)^2 \\
&= \frac{(1 + L_y^2)L_x^2 - 2L_xL_y(L_xL_y) + (1 + L_x^2)L_y^2}{1 + L_x^2 + L_y^2} \\
&= \frac{L_x^2 + L_y^2}{1 + L_x^2 + L_y^2} \\
&= \frac{|\nabla L|^2}{1 + |\nabla L|^2}
\end{aligned} \tag{C.15}$$

where  $|\nabla L|^2 = L_x^2 + L_y^2$ , the Euclidean gradient magnitude square. If  $f(x, y) = \alpha L(x, y)$ , i.e., a scaled function of  $L$ , we have:

$$|\text{grad}(\alpha L)|^2 = \frac{\alpha^2 |\nabla L|^2}{1 + \alpha^2 |\nabla L|^2} = \frac{|\nabla L|^2}{\alpha^{-2} + |\nabla L|^2} \tag{C.16}$$

## D Online Demo System Implementation

In designing the online classification system, we face the following challenges:

1. Processing speed: For user-friendliness, the system should not take too long (e.g., a few minutes) for processing a submitted image. The submitted images can be of various size and the processing time in general depends on the image size. Therefore, reducing the processed image size (e.g., by resizing or central-region cropping) is a strategy for improving the processing speed.
2. Classification accuracy: For the usefulness of the system, the classification should have a reasonably good classification accuracy as compared to random guessing. However, as reduction in the processed image size may degrade the classification accuracy, the criteria of processing speed and classification accuracy are in a tradeoff relationship. Therefore, we need to find a strategy

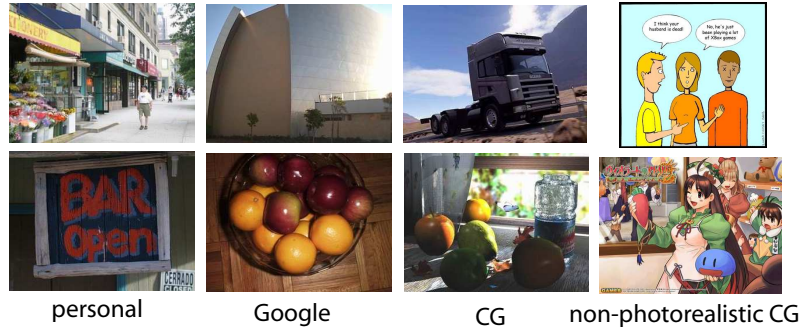


Figure D.1: Example images from the four categories of the online classifier training dataset.

to improve the classification accuracy for the case of reduced-size images.

3. The diversity of the input images: It is not too surprising that an online system would encounter input images of diverse types, which includes PRCG, non-photorealistic CG, CG-and-PIM-hybrid images, painting or drawing and so on. However, most of the experiments presented in the related paper [39, 55, 70] do not consider such a wide spectrum of images. In order to obtain a well-performing classifier, we need to include images of wide diversity into the training dataset.

### D.1 Dataset with Non-photorealistic Computer Graphics

In [70], the experiments are performed on the *Columbia Photographic Images and Photorealistic Computer Graphics Dataset* [69] which includes only photorealistic computer graphics. One of the challenges in system design is the potential wide diversity of the input images and that non-photorealistic CG images are actually very common on the Internet. In order to improve the classification performance, we have to include another set of 800 non-photorealistic computer graphics (NPRCG) into the online classifier training dataset. The NPRCG includes cartoon, drawing, 2D graphics, images of presentation slide, logo images and so on. These images are

collected from the Google Image Search. As a result, our training image set consists of four categories as shown in Figure D.1. Note that in training the classifiers used for this online system, the recaptured CG category in the Columbia Photographic Images and Photorealistic Computer Graphics Dataset is excluded as the recaptured CG category mainly caters for the recapturing attack experiment.

## D.2 Image Downsizing

For addressing the system design challenge of processing speed, we downsize the input images such that the longer dimension of the downsized image is of 360 pixels (the entire content of an image is retained, so is the aspect ratio). As the algorithm for extracting the features is proportional to the image size, the reduction of computational load due to image downsizing is substantial (about more than two times in average) as the average size of the web images is around  $700 \times 500$  pixels.

In a prior work [55], computational efficiency is obtained by cropping out the smaller-size central area from the images. In order to evaluate the two strategies of reducing image size, we train support vector machines (SVM) of the LIBSVM [37] implementation using the dataset mentioned in Section D.1 for separate cases where the images are downsized, central-cropped and with the original size. In this case, the *Personal* and *Google* categories form a class, while the *PRCG* and *NPRCG* categories form the opposite class. We use the radial basis function (RBF) kernel for the SVM and model selection (for the regularization and the kernel parameters) is done by a grid search [37] in the joint parameter space. The classification performance we report hereupon is based on a five-fold cross-validation procedure.

Table D.1 shows the SVM classification accuracy corresponding to the two different image size reduction strategies, i.e., downsizing and central cropping, while comparing to the SVM performance when there is no reduction of image size. Note

Table D.1: SVM Classification Accuracy for Different Image Size Reduction Strategies

Classifier	Original size	Downsizing	Central Cropping
Geometry	83.8%	78.2%	79.9%
Wavelets	81.2%	77.3%	72.8%
Cartoon	76.1%	73.1%	75.9%

that central cropping results in a slightly better classification accuracy for the geometry (+1.7%) and the cartoon (+2.8%) classifier, but causes a serious drop of performance for the wavelet ( $-4.5\%$ ) classifier. In this case, it is inconclusive on which image reduction strategy is better, but the degradation of the classifier performance due to image downsizing is more uniform over all the classifiers.

### D.3 Classifier Fusion

The image downsizing results in an average of 4.2% decrease (over the three classifiers) in the classification accuracy when comparing to the case of no image size reduction. To counter the performance decrease, we consider fusing the SVM classification outputs from the geometry, the wavelet and the cartoon classifiers. The effort of fusing a set of base classifiers is only justified when the fusion classifier outperforms the best among the base classifiers. According to an analysis [35], this only happens when the individual base classifiers are reasonably accurate, individual performances are comparable, and their errors are uncorrelated. Although these conditions may be not satisfied in practice, fusion of classifiers often leads to a better classification performance and three fundamental reasons for the good performance are proposed [15]. Firstly, in the case of a small dataset, different classifiers may provide different but equally good hypothesis. Fusion of these hypothesis reduces the risk of choosing the wrong hypothesis. Secondly, for learning algorithms which could only achieve local optima, fusion of hypothesis corresponding to the multiple

local optima may be a better approximation to the target function. Thirdly, for the case where the target function is not in the hypothesis space of the individual base classifiers, the fused classifier may have an enhanced hypothesis space which includes the target function.

We evaluate three fusion strategies, the normalized ensemble fusion [95], the score concatenation fusion, the majority vote fusion. The normalized ensemble fusion procedure consists of three steps:

1. **Normalization of the distance of the data points to the SVM decision boundary:** The paper suggests three types of normalization schemes, i.e., rank normalization, range normalization and Gaussian normalization, for producing normalized scores.
2. **Combination of the normalized scores:** The paper suggests functions such as minimum, maximum, average, product, inverse entropy and inverse variance for combining the normalized scores.
3. **Finding the optimal fusion strategy:** The optimal fusion strategy is obtained by searching over all the normalization schemes and the combination functions.

The features obtained from the normalized ensemble fusion are then used for training the final fusion SVM.

Before performing the score concatenation fusion and the majority vote fusion, the binary SVM output is mapped into posterior probability by fitting the empirical posterior histogram of the distance of the data points to the decision boundary using



Table D.2: Downsized Image Classification Accuracy for Different Fusion Strategies

Normalized ensemble	Score concatenation	Majority vote	Best base classifier (geometry)
79.7%	80.0%	77.4%	78.2%

a parametric sigmoid function [77]:

$$P(y = 1|f) = \frac{1}{1 + \exp(Af + B)} \tag{D.17}$$

where  $f$  is the distance from the decision boundary and  $(A, B)$  are the sigmoid model parameters. The score concatenation fusion concatenates the posterior probability of all the base classifiers and the concatenated features are used for training the final SVM. The majority vote fusion selects the final decision by a majority vote from the base classifiers. In our evaluation, we compute the classification accuracy for the three above-mentioned fusion strategies, using the dataset of downsized images. The classification accuracy is as shown in Table D.2. Note the accuracy reported here is for classifying downsized images. From the results, we see the use of classifier fusion indeed improves the accuracy in classifying downsized images by about 1.8%, as compared to the best performing base (geometry) classifier.

Note that the normalized ensemble fusion and the score concatenation fusion perform equally well on the dataset, while the the majority vote fusion result seems to be just about the average of the classification accuracies of the three base classifiers. Due to the simplicity of the score concatenation fusion as compared to the normalized ensemble fusion, we choose to use the score concatenation fusion method.

#### D.4 Exploiting Dataset Heterogeneity

The increase of the classification accuracy for the score concatenation fusion is only 1.8%, which is quite minor. To further improve the classification accuracy for downsized images, we aim to find a better way to generate a set of base classifiers by exploiting the diversity and heterogeneity within the dataset. As we can see that while the *PRCG* and the *NPRCG* image categories form a single class (CG), they are actually two very different types of images. Similarly, the *Personal* and the *Google* image categories are different in the sense that the *Personal* category consists of the typical camera images from a few high-end cameras with diverse content while the *Google* category contains images from diverse models of camera potentially having undergone various types of additional post-processing.

From the dataset, we can generate nine (i.e.,  $3^2$ ) sets of two-class data by exhaustively combining the elements of the power set of the two classes, as shown in Figure D.2. We train a SVM base classifier for each of the two-class data subset combinations. The binary outputs of the SVM base classifiers are mapped to posterior probability by fitting a sigmoid function, as given in Equation D.17. Then, the posterior probability for all base classifiers are combined through the score concatenation fusion for training the final fusion SVM classifier.

The accuracy of the fusion classifiers for the dataset of original size images and downsized images are shown in Table D.3. The above approach is conceptually similar to a common machine learning technique, called bagging, in which different subsets of data are used to train multiple classifiers to be combined later. Each individual classifier has the potential to capture the differences between classes contained in each unique subset of data. The difference of our implementation from the conventional bagging is that in our case subsets of data are obtained according to the

Table D.3: Classification Accuracy After Considering Dataset Heterogeneity

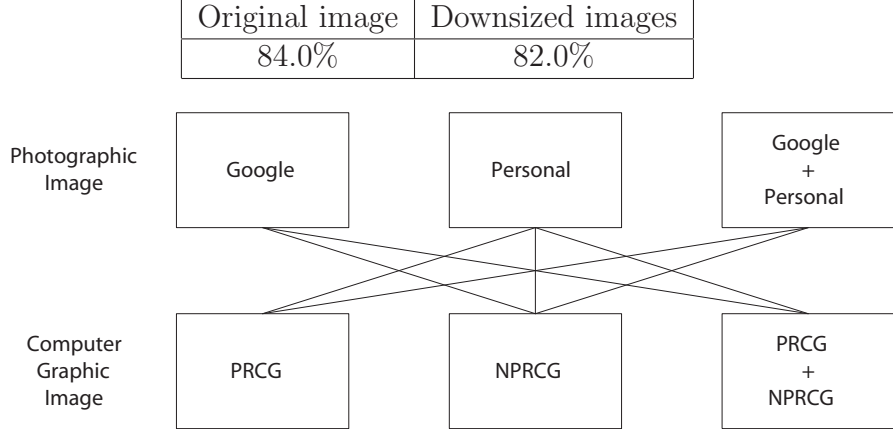


Figure D.2: Nine combinations of the data subsets.

data subtypes, rather than sampling of the whole set. Note that for the downsized images, the gain in classification accuracy after considering dataset heterogeneity is 2%, while there is very little gain for the original size images (84.0% compared to best performing geometry classifier with a classification accuracy of 83.8%). To explain the observation, we conjecture that image downsizing has made the base classifiers less stable and decreased the correlation of error between the base classifiers. Therefore, according the analysis mentioned before, there is a greater chance for classifier fusion to be effective in the case of downsized images.

In summary, we address the system design challenges by reducing the processed image size, including an additional set of non-photorealistic computer graphic images into the training dataset, and exploiting the heterogeneity within the dataset. By adopting this strategy, the per-image processing time is reduced by more than two times, the classification accuracy degrades only by 2.0% as compared to the case with no image downsizing, and the system can now handle non-photorealistic computer graphic images.

## E Proof of the Integral Solution to CRF Property (Property 5)

The partial differential equation (PDE) in Eq. 4.4

$$\mathcal{G}_1(R) = \frac{f''(f^{-1}(R))}{(f'(f^{-1}(R)))^2} \quad (\text{E.18})$$

can be rewritten as:

$$\mathcal{G}_1(R) = \frac{d}{dR} (\ln f'(f^{-1}(R))) \quad (\text{E.19})$$

Then, we can solve for the function  $f^{-1}$  as below:

$$\ln f'(f^{-1}(R)) = \int \mathcal{G}_1(R) dR \quad (\text{E.20})$$

$$f'(f^{-1}(R)) = \exp\left(\int \mathcal{G}_1(R) dR\right) \quad (\text{E.21})$$

Let  $r = f^{-1}(R)$ , then, we have:

$$f'(f^{-1}(R)) = \frac{dR}{dr} = \exp\left(\int \mathcal{G}_1(R) dR\right) \quad (\text{E.22})$$

$$\frac{dr}{dR} = \exp\left(-\int \mathcal{G}_1(R) dR\right) \quad (\text{E.23})$$

$$f^{-1}(R) = \int \exp\left(-\int \mathcal{G}_1(R) dR\right) dR \quad (\text{E.24})$$

## F Proof of the Decomposition of $\mathcal{G}_1$ Proposition (Proposition 3)

In a general 2D  $(u_1, u_2)$ -Cartesian coordinate frame, let's denote  $R_{u_1}$  and  $R_{u_2}$ , the 1st-order partial derivatives of a function  $R(u_1, u_2)$ , respectively as  $R_1$  and  $R_2$ , and we follow the similar notation for the 2nd-order partial derivatives.

The vector  $(R_1, R_2)$  forms a 1-tensor, which transforms to new vector  $(R'_1, R'_2)$  under a rotation, according to the tensorial transformation law:

$$\begin{pmatrix} R'_1 \\ R'_2 \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} R_1 \\ R_2 \end{pmatrix} = T \begin{pmatrix} R_1 \\ R_2 \end{pmatrix} \quad (\text{F.25})$$

where  $T$  is a  $2 \times 2$  rotation matrix and  $\alpha$  is the rotation angle. The above transformation equation can be more concisely written as below where the repeated index are summed over all possible substitutions:

$$R'_i = T_{ij}R_j \rightarrow \begin{cases} R'_1 = T_{11}R_1 + T_{12}R_2 \\ R'_2 = T_{21}R_1 + T_{22}R_2 \end{cases} \quad (\text{F.26})$$

On the other hand, the Hessian is a 2-tensor, which transforms according to the tensorial rule below under a rotation:

$$R'_{ij} = T_{ik}T_{jl}R_{kl} \quad (\text{F.27})$$

With the above two transformation rules, we can easily express the geometry invariant quantities in  $(u_{\bar{t}}, u_{\bar{g}})$  coordinates with the derivative quantities in  $(u_t, u_g)$  coordinates (these two coordinate frames are shown in Fig. 4.2). In this case, the rotation from the  $(u_t, u_g)$ -coordinate frame to the  $(u_{\bar{t}}, u_{\bar{g}})$ -coordinate frame is given

by the rotation matrix:

$$T = \begin{pmatrix} \cos(\frac{\pi}{4}) & \sin(\frac{\pi}{4}) \\ -\sin(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \quad (\text{F.28})$$

The expressions in Proposition 3 is given as below. Note that, the first-order derivative in the tangential direction,  $R_t = 0$ .

$$\frac{R_{\bar{t}\bar{t}}}{R_{\bar{t}}^2} = \frac{T_{\bar{t}k}T_{\bar{t}l}R_{kl}}{(T_{\bar{t}j}R_j)^2} \quad (\text{F.29})$$

$$= \frac{T_{\bar{t}\bar{t}}T_{\bar{t}\bar{t}}R_{\bar{t}\bar{t}} + 2T_{\bar{t}\bar{t}}T_{\bar{t}g}R_{\bar{t}g} + T_{\bar{t}g}T_{\bar{t}g}R_{gg}}{(T_{\bar{t}\bar{t}}R_{\bar{t}} + T_{\bar{t}g}R_g)^2} \quad (\text{F.30})$$

$$= \frac{\frac{1}{2}R_{\bar{t}\bar{t}} + R_{\bar{t}g} + \frac{1}{2}R_{gg}}{\frac{1}{2}R_g^2} \quad (\text{F.31})$$

$$= \frac{\lambda - \kappa - 2\mu}{R_g} \quad (\text{F.32})$$

$$\frac{R_{\bar{g}\bar{g}}}{R_{\bar{g}}^2} = \frac{T_{\bar{g}k}T_{\bar{g}l}R_{kl}}{(T_{\bar{g}j}R_j)^2} \quad (\text{F.33})$$

$$= \frac{T_{\bar{g}\bar{t}}T_{\bar{g}\bar{t}}R_{\bar{t}\bar{t}} - 2T_{\bar{g}\bar{t}}T_{\bar{g}g}R_{\bar{t}g} + T_{\bar{g}g}T_{\bar{g}g}R_{gg}}{(T_{\bar{g}\bar{t}}R_{\bar{t}} + T_{\bar{g}g}R_g)^2} \quad (\text{F.34})$$

$$= \frac{\frac{1}{2}R_{\bar{t}\bar{t}} - R_{\bar{t}g} + \frac{1}{2}R_{gg}}{\frac{1}{2}R_g^2} \quad (\text{F.35})$$

$$= \frac{\lambda - \kappa + 2\mu}{R_g} \quad (\text{F.36})$$

$$\frac{R_{\bar{t}\bar{g}}}{R_{\bar{t}}R_{\bar{g}}} = \frac{T_{\bar{t}k}T_{\bar{g}l}R_{kl}}{T_{\bar{t}i}R_iT_{\bar{g}j}R_j} \quad (\text{F.37})$$

$$= \frac{T_{\bar{t}t}T_{\bar{g}t}R_{tt} - 2T_{\bar{t}t}T_{\bar{g}g}R_{tg} + T_{\bar{t}g}T_{\bar{g}g}R_{gg}}{(T_{\bar{t}t}R_t + T_{\bar{t}g}R_g)(T_{\bar{g}t}R_t + T_{\bar{g}g}R_g)} \quad (\text{F.38})$$

$$= \frac{-\frac{1}{2}R_{tt} + \frac{1}{2}R_{gg}}{\frac{1}{2}R_g^2} \quad (\text{F.39})$$

$$= \frac{\lambda + \kappa}{R_g} \quad (\text{F.40})$$

## G Proof of the Geometric Significance of Equality Constraint Proposition (Proposition 4)

$$\left( \frac{R_{\bar{t}\bar{t}}}{R_{\bar{t}}^2} = \frac{R_{\bar{g}\bar{g}}}{R_{\bar{g}}^2} = \frac{R_{\bar{t}\bar{g}}}{R_{\bar{t}}R_{\bar{g}}} \right) \quad (\text{G.41})$$

$$\iff \left( \frac{R_{\bar{t}\bar{t}}}{R_{\bar{t}}^2} - \frac{R_{\bar{g}\bar{g}}}{R_{\bar{g}}^2} = 0 \right) \& \left( \frac{R_{\bar{t}\bar{t}}}{R_{\bar{t}}^2} - \frac{R_{\bar{t}\bar{g}}}{R_{\bar{t}}R_{\bar{g}}} = 0 \right) \quad (\text{G.42})$$

$$\iff (\mu = 0) \& (\mu + \kappa = 0) \quad (\text{G.43})$$

$$\iff (\kappa = \mu = 0) \quad (\text{G.44})$$

Futhermore,

$$(\kappa = \mu = 0) \iff (R_{tt} = R_{tg} = 0) \quad (\text{G.45})$$

$R_{tg} = 0$  implies that the function  $R(u_t, u_g)$  is in the form of either  $a(u_t)$  or  $b(u_g)$ , where  $a(u_t)$  and  $b(u_g)$  are respectively a single parameter function of  $u_t$  and  $u_g$ . Then, with  $R_{tt} = 0$ , it implies that  $R(u_t, u_g)$  must take the form of  $b(u_g)$ .

## H Proof of the Error Metric Calibration Proposition (Proposition 5)

The purpose of calibrating the error metric for  $Q(R)$  is so that the calibrated error metric would be linear to the error metric in the CRF space. The calibration is desirable as it makes the optimization in the  $Q(R)$  space equivalent to an optimization in the CRF space, when using the common least square optimization criterion. The calibration of error metric for  $Q(R)$  can be formulated as reparametrization of a one-dimensional curve in a  $n$ -dimensional space.

Assume that the image irradiance value  $r \in [0, 1]$  is uniformly sampled at  $N$  points and a sample point is denoted as  $r_i$ . For gamma curve, the relationship between image intensity  $R_i$  and image irradiance  $r_i$  is given by:

$$R_i(\gamma) = f(r_i) = r_i^\gamma \quad (\text{H.46})$$

In the CRF space, the error metric is measured by the root mean squared error (RMSE):

$$RMSE(f_1, f_2) = \left( \frac{1}{N} \sum_{i=1}^N (f_1(r_i) - f_2(r_i))^2 \right)^{0.5} \quad (\text{H.47})$$

From Property 6,  $Q(R) = \gamma$  for a gamma curve  $f(r) = r^\gamma$ . Then, the RMSE error metric between two  $Q(R)$  functions,  $Q_1(R)$  and  $Q_2(R)$  (respectively corresponds to gamma curves  $f_1 = r_1^\gamma$  and  $f_2 = r_2^\gamma$ ), is given by:

$$RMSE(Q_1, Q_2) = \left( \frac{1}{N} \sum_{i=1}^N (Q_1(R_i) - Q_2(R_i))^2 \right)^{0.5} \quad (\text{H.48})$$

$$= |\gamma_1 - \gamma_2| \quad (\text{H.49})$$



Note that the set of values  $\{R_i\}$  can be considered as a point  $[R_1, R_2, \dots, R_N]$  in an  $N$ -dimensional space. As the parameter  $\gamma$  changes, it will trace out a one-dimensional curve in the  $N$ -dimensional space. For a differential change for  $\gamma$ , the differential change for  $R_i$  is given by:

$$dR_i = r_i^\gamma \ln(r_i) d\gamma \quad (\text{H.50})$$

Then, we can define the differential change for the arc length  $dS$  of the 1D space curve as the RMSE given in Eq. H.47, which is rewritten as below:

$$dS = \left( \frac{1}{N} \sum_{i=1}^N dR_i^2 \right)^{\frac{1}{2}} = \left( \frac{1}{N} \sum_{i=1}^N (r_i^\gamma \ln(r_i) d\gamma)^2 \right)^{\frac{1}{2}} \quad (\text{H.51})$$

When we let the uniform sampling grid on  $r \in [0, 1]$  to become infinitely fine, we are sending the number of sampling points  $N$  on  $r$  to infinity. Then, the differential change for the arc length  $dS$  becomes:

$$dS = \lim_{N \rightarrow \infty} \left( \frac{1}{N} \sum_{i=1}^N dR_i^2 \right)^{\frac{1}{2}} = \left( \int_0^1 (r^\gamma \ln(r))^2 dr \right)^{\frac{1}{2}} d\gamma \quad (\text{H.52})$$

Note that the differential change for the arc length  $dS$  represents the error metric in the CRF space, and the differential change  $dQ = |d\gamma|$  represents the error metric in the  $Q(R)$  space (Once we choose a proper direction to trace the space curve, we have  $|d\gamma| = d\gamma$ , where the sign is no longer an issue). We can see that

$$\frac{dS}{dQ} = \left( \int_0^1 (r^\gamma \ln(r))^2 dr \right)^{\frac{1}{2}} \quad (\text{H.53})$$

is a function of  $\gamma$ , therefore the same differential distances from two different  $Q(R)$

functions in the  $Q(R)$  space may correspond to two different differential distances in the CRF space. In the differential geometry context, we say that the  $Q(R)$  space is not flat. A drawback of such non-flatness is that when we perform curve-fitting using the least-square optimization criterion in the  $Q(R)$  space, the same optimization cost measured at two functions  $Q_a(R)$  and  $Q_b(R)$  may corresponds to two different error measurements in the CRF space. This is a bad news because what we are eventually interested is to minimize the error in the CRF space. We can remove the non-flatness by reparametrizing the space curve with its arc length, so that after the reparametrization we have  $\frac{dS}{dQ} = \text{constant}$ , where a transformation of  $Q(R)$  to  $\bar{Q}(R)$  is resulted by the arc length reparametrization. For the reparametrization, we need to compute the arc length by solving the integration in Eq. H.52, which gives us:

$$S = -\sqrt{\frac{2}{2\gamma + 1}} + C \quad (\text{H.54})$$

where  $C$  is a constant. As we have  $Q(R) = \gamma$  for the gamma curves, we can transform the  $Q(R)$  linearly according to Eq. H.54 as if it is  $\gamma$ :

$$\bar{Q}(R) = \beta(Q) = k\sqrt{\frac{2}{2Q(R) + 1}} + C \quad (\text{H.55})$$

where  $k$  and  $C$  are constant. By doing so, we achieve our goal for  $Q(R)$  error metric calibration (equivalently, the flattening of the space curve metric) such that  $\frac{dS}{dQ} = \text{constant}$ .

To determine the constants  $k$  and  $C$ , we import boundary conditions:  $\beta(0) = 0$  and  $\beta(1) = 1$ . This condition is on the assumption that the gamma curve are convex, i.e.,  $\gamma \in [0, 1]$ . With these boundary conditions, we finally obtain the expression

given in Proposition 5:

$$\bar{Q} = \frac{\sqrt{3}}{\sqrt{3}-1} \left( 1 - \sqrt{\frac{1}{2Q+1}} \right) \quad (\text{H.56})$$

## I The General Expression for Geometry Invariant Computation

The isophote curvature ( $\kappa$ ), the normalized 2nd-derivative in the gradient direction ( $\lambda$ ), and the flow line curvature ( $\mu$ ) given in Eq. I.57 are geometric quantities of the image intensity function  $R$  and have nothing to do with the coordinate frames on which they are computed.

$$\kappa = -\frac{R_{tt}}{R_g}, \quad \lambda = \frac{R_{gg}}{R_g} \quad \text{and} \quad \mu = -\frac{R_{tg}}{R_g} \quad (\text{I.57})$$

Therefore, these quantities can be expressed in a general expression for which computation can be done on any coordinate frame as below [23]. The expressions below are all given in the manifest index notation for which the repeated index are summed over all possible substitutions.

$$\kappa = \frac{R_i \epsilon_{ij} R_{jk} \epsilon_{kl} R_l}{(R_m R_m)^{\frac{3}{2}}} \quad (\text{I.58})$$

$$\mu = \frac{R_i \epsilon_{ij} R_{jk} \delta_{kl} R_l}{(R_m R_m)^{\frac{3}{2}}} \quad (\text{I.59})$$

$$\lambda = \frac{R_i \delta_{ij} R_{jk} \delta_{kl} R_l}{(R_m R_m)^{\frac{3}{2}}} \quad (\text{I.60})$$

where  $\delta_{ij}$  is the symmetric Kronecker tensor, and  $\epsilon_{ij}$  is the anti-symmetric Levi-Civita tensor, as defined below:

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \quad (\text{I.61})$$

$$\epsilon_{ij} = \begin{cases} 1 & \text{if } (i, j) \text{ is even} \\ -1 & \text{if } (i, j) \text{ is odd} \\ 0 & \text{otherwise} \end{cases} \quad (\text{I.62})$$

For the definition of  $\epsilon_{ij}$ , an index sequence  $(i, j)$  is called even if an even number of pairwise swapping of indexes is needed to restore it back to an ordered sequence. An odd index sequence is similarly defined. For expressions in Eq. I.58, Eq. I.59, and Eq. I.60, the indexes can be replaced by the coordinate index. For instance, with the original  $(x, y)$ -coordinate frame,  $\kappa$ ,  $\mu$ , and  $\lambda$  can be written as:

$$\kappa = \frac{2R_x R_y R_{xy} - R_x^2 R_{yy} - R_y^2 R_{xx}}{(R_x^2 + R_y^2)^{\frac{3}{2}}} \quad (\text{I.63})$$

$$\mu = \frac{(R_x^2 - R_y^2)R_{xy} + R_x R_y (R_{yy} - R_{xx})}{(R_x^2 + R_y^2)^{\frac{3}{2}}} \quad (\text{I.64})$$

$$\lambda = \frac{R_x^2 R_{xx} + 2R_x R_y R_{xy} R_y^2 R_{yy}}{(R_x^2 + R_y^2)^{\frac{3}{2}}} \quad (\text{I.65})$$

Finally, by substituting Eq. I.63, Eq. I.64, and Eq. I.65 into the expression in Eq. 4.13, we obtain:

$$\frac{R_{\bar{g}\bar{g}}}{R_{\bar{g}}^2} = \frac{R_x^2(\Delta R - 2R_{xy}) + R_y^2(\Delta R + 2R_{xy}) + 2R_x R_y \bar{\Delta} R}{(R_x^2 + R_y^2)^2} \quad (\text{I.66})$$

$$\frac{R_{\bar{t}\bar{t}}}{R_{\bar{t}}^2} = \frac{R_x^2(\Delta R + 2R_{xy}) + R_y^2(\Delta R - 2R_{xy}) - 2R_x R_y \bar{\Delta} R}{(R_x^2 + R_y^2)^2} \quad (\text{I.67})$$

$$\frac{R_{\bar{g}\bar{t}}}{R_{\bar{g}} R_{\bar{t}}} = \frac{(R_x^2 + R_y^2) \bar{\Delta} R + 4R_x R_y R_{xy}}{(R_x^2 + R_y^2)^2} \quad (\text{I.68})$$

where  $\Delta R = R_{xx} + R_{yy}$  and  $\bar{\Delta} R = R_{xx} - R_{yy}$ .

# Bibliography

- [1] Akenine-Moller, T., Moller, T., and Haines, E. (2002). *Real-Time Rendering*. A. K. Peters, Ltd., MA.
- [2] Amsberry, C. (1989). Alterations of photos raise host of legal, ethical issues. *The Wall Street Journal*.
- [3] Athitsos, V., Swain, M., and Frankel, C. (1997). Distinguishing photographs and graphics on the world wide web. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 10–17.
- [4] Avcibas, I., Bayram, S., Memon, N., Ramkumar, M., and Sankur, B. (2004). A classifier design for detecting image manipulations. In *IEEE International Conference on Image Processing*, volume 4, pages 2645 – 2648, Singapore.
- [5] Bertalmio, M., Bertozzi, A., and Sapiro, G. (2001). Navier-stokes, fluid dynamics, and image and video inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- [6] Bertalmio, M., Sapiro, G., Caselles, V., and Ballester, C. (2000). Image inpainting. In *ACM SIGGRAPH*, pages 417–424.
- [7] Bhattacharjee, S. and Kutter, M. (1998). Compression tolerant image authentication. In *IEEE International Conference on Image Processing*.

- [8] Calphoto (2000). A database of photos of plants, animals, habitats and other natural history subjects.
- [9] Chang, E. C., Kankanhalli, M. S., Guan, X., Huang, Z., and Wu, Y. (2003). Robust image authentication using content based compression. *Multimedia Systems*, 9(2):121–130.
- [10] Chen, W., Shi, Y. Q., and Wei, S. (2007). Image splicing detection using 2-D phase congruency and statistical moments of characteristic function. In *SPIE Electronic Imaging*, San Jose, CA.
- [11] Cover, T. M. and Thomas, J. A. (1991). *Elements of information theory*. Wiley-Interscience, New York, NY, USA.
- [12] Cox, I., Bloom, J., and Miller, M. (2001). *Digital Watermarking, Principles and Practice*. Morgan Kaufmann.
- [13] de Boor, C. (1978). *A Practical Guide to Splines*. Springer-Verlag, New York.
- [14] Debevec, P. E. and Malik, J. (1997). Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH*, pages 369–378.
- [15] Dietterich, T. G. (2000). Ensemble methods in machine learning. *Lecture Notes in Computer Science*, 1857:1–15.
- [16] Efros, A. A. and Freeman, W. T. (2001). Image quilting for texture synthesis and transfer. In *ACM SIGGRAPH*, pages 341–346.
- [17] Fackrell, J. W. A. and McLaughlin, S. (1994). The higher-order statistics of speech signals. In *IEE Colloquium on Techniques for Speech Processing and their Application*, page 7.

- [18] Farid, H. (1999). Detecting digital forgeries using bispectral analysis. MIT AI Memo AIM-1657, MIT.
- [19] Farid, H. (2001). Blind inverse gamma correction. *IEEE Transactions on Image Processing*, 10(10):1428–1433.
- [20] Farid, H. (2006). Digital image ballistics from JPEG quantization. Technical Report TR2006-583, Department of Computer Science, Dartmouth College.
- [21] Farid, H. and Lyu, S. (2003). Higher-order wavelet statistics and their application to digital forensics. In *IEEE Workshop on Statistical Analysis in Computer Vision*, Madison, Wisconsin.
- [22] Field, D. J. (1987). Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, 4(12):2379–2394.
- [23] Florack, L. M. J., ter Haar Romeny, B. M., Koenderink, J. J., and Viergever, M. A. (1992). Scale and the differential structure of images. *Image Vision Comput.*, 10(6):376–388.
- [24] Freeman, W. T., Pasztor, E. C., and Carmichael, O. T. (2000). Learning low-level vision. *International Journal of Computer Vision*, 40(1):25–47.
- [25] Fridrich, J. (1998). Image watermarking for tamper detection. In *IEEE International Conference on Image Processing*, volume 2.
- [26] Fridrich, J., Goljan, M., and Baldoza, A. C. (2000). New fragile authentication watermark for images. In *IEEE International Conference on Image Processing*, volume 1.



- [27] Fridrich, J., Soukal, D., and Lukas, J. (2003). Detection of copy-move forgery in digital images. In *Digital Forensic Research Workshop*, Cleveland, OH.
- [28] Friedman, G. L. (1993). The trustworthy digital camera: restoring credibility to the photographic image. *IEEE Transactions on Consumer Electronics*, 39(4):905–910.
- [29] Friedman, T. L. (2006). *The world is flat: the globalized world in the twenty-first century*. Penguin Books, London.
- [30] Fu, D., Shi, Y. Q., and Su, W. (2006). Detection of image splicing based on hilbert-huang transform and moments of characteristic functions with wavelet decomposition. In *International Workshop on Digital Watermarking*, Jeju, Korea.
- [31] Fu, D., Shi, Y. Q., and Su, W. (2007). A generalized Benford’s law for JPEG coefficients and its applications in image forensics. In *SPIE Electronic Imaging*, San Jose, CA.
- [32] Gonzalez, R. and Woods, R. (1987). *Digital image processing*. Addison-Wesley Reading, Mass.
- [33] Gool, L. J. V., Moons, T., and Ungureanu, D. (1996). Affine/photometric invariants for planar intensity patterns. In *European Conference on Computer Vision*, pages 642–651.
- [34] Grossberg, M. and Nayar, S. (2003). What is the space of camera response functions? In *IEEE Computer Vision and Pattern Recognition*, pages 602–609.
- [35] Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 12(10):993–1001.

- [36] He, J., Lin, Z., Wang, L., and Tang, X. (2006). Detecting doctored JPEG images via DCT coefficient analysis. In *European Conference on Computer Vision*, volume 3953.
- [37] Hsu, C.-W., Chang, C.-C., and Lin, C.-J. (2003). A practical guide to support vector classification.
- [38] Hsu, Y.-F. and Chang, S.-F. (2006). Detecting image splicing using geometry invariants and camera characteristics consistency. In *International Conference on Multimedia and Expo (ICME)*, Toronto, Canada.
- [39] Ianeva, T., de Vries, A., and Rohrig, H. (2003). Detecting cartoons: A case study in automatic video-genre classification. In *IEEE International Conference on Multimedia and Expo*, volume 1, pages 449–452.
- [40] Johnson, M. and Farid, H. (2005). Exposing digital forgeries by detecting inconsistencies in lighting. In *ACM Multimedia and Security Workshop*, New York, NY.
- [41] Kim, Y. C. and Powers, E. J. (1979). Digital bispectral analysis and its applications to nonlinear wave interactions. *IEEE Trans. on Plasma Science*, (2):120–131.
- [42] Krieger, G., Zetsche, C., and Barth, C. (1997). Higher-order statistics of natural images and their exploitation by operators selective to intrinsic dimensionality. In *IEEE Signal Processing Workshop on Higher-Order Statistics*, Washington, DC, USA.
- [43] Lee, A. B., Pedersen, K. S., and Mumford, D. (2003). The nonlinear statistics

- of high-contrast patches in natural images. *International Journal of Computer Vision*, 54(1):83–103.
- [44] Lee, J. (1997). *Riemannian Manifolds: an introduction to curvature*. Springer Verlag.
- [45] Lee, J. (2002). *Introduction to Smooth Manifolds*. Springer.
- [46] Lin, C.-Y. and Chang, S.-F. (1998). A robust image authentication method surviving JPEG lossy compression. In *SPIE Storage and Retrieval of Image/Video Database*, volume 3312.
- [47] Lin, C.-Y. and Chang, S.-F. (2001). A robust image authentication method distinguishing JPEGcompression from malicious manipulation. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(2):153–168.
- [48] Lin, E. T., Podilchuk, C. I., and Delp, E. J. (2000). Detection of image alterations using semi-fragile watermarks. In *SPIE International Conference on Security and Watermarking of Multimedia Contents II*, volume 3971, pages 152–163.
- [49] Lin, S., Gu, J., Yamazaki, S., and Shum, H.-Y. (2004). Radiometric calibration from a single image. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 938–945.
- [50] Lin, S. and Zhang, L. (2005). Determining the radiometric response function from a single grayscale image. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 66–73.
- [51] Lin, Z., Wang, R., Tang, X., and Shum, H.-Y. (2005). Detecting doctored images using camera response normality and consistency. In *IEEE Computer*

- Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1087–1092.
- [52] Lukas, J. and Fridrich, J. (2003). Estimation of primary quantization matrix in double compressed JPEG images. In *Digital Forensic Research Workshop*.
- [53] Lukas, J., Fridrich, J., and Goljan, M. (2006). Detecting digital image forgeries using sensor pattern noise. In *SPIE Electronic Imaging, Photonics West*.
- [54] Luo, W., Huang, J., and Qiu, G. (2006). Robust detection of region-duplication forgery in digital image. *International Conference on Pattern Recognition*.
- [55] Lyu, S. and Farid, H. (2005). How realistic is photorealistic? *IEEE Transactions on Signal Processing*, 53(2):845–850.
- [56] Mahajan, D., Ramamoorthi, R., and Curless, B. (2006). A theory of spherical harmonic identities for BRDF/lighting transfer and image consistency. In *European Conference on Computer Vision*, Graz, Austria.
- [57] Mandelbrot, B. B. (1983). *The fractal geometry of nature*. San Francisco: W.H. Freeman.
- [58] Mann, S. (2000). Comparametric equations with practical applications in quantigraphic image processing. *IEEE Trans. Image Proc.*, 9(8):1389–1406.
- [59] Meer, P. and Weiss, I. (1990). Smoothed differentiation filters for images. In *International Conference on Pattern Recognition*, pages 121–126, Atlantic City, NJ.
- [60] Memon, N. and Vora, P. (1999). Authentication techniques for multimedia content. In *SPIE Multimedia Systems and Applications*, volume 3528, pages 412–422.

- [61] Memon, N., Vora, P., Yeo, B. L., and Yeung, M. (2000). Distortion bounded authentication techniques. In *SPIE International Conference on Security and Watermarking of Multimedia Contents II*, volume 3971, pages 164–174.
- [62] Meriam, J. L. and Kraige, L. G. (1986). *Engineering Mechanics Volume 2: Dynamics*. John Wiley and Sons, New York.
- [63] Meyer, G. W., Rushmeier, H. E., Cohen, M. F., Greenberg, D. P., and Torrance, K. E. (1986). An experimental evaluation of computer graphics imagery. *ACM SIGGRAPH*, 5(1):30–50.
- [64] Mitsunaga, T. and Nayar, S. (1999). Radiometric self calibration. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 374–380.
- [65] Ng, T.-T. and Chang, S.-F. (2004a). Classifying photographic and photorealistic computer graphic images using natural image statistics. ADVENT Technical Report 220-2006-6, Columbia University.
- [66] Ng, T.-T. and Chang, S.-F. (2004b). A data set of authentic and spliced image blocks. ADVENT Technical Report 203-2004-3, Columbia University.
- [67] Ng, T.-T. and Chang, S.-F. (2004c). A model for image splicing. In *IEEE International Conference on Image Processing*, Singapore.
- [68] Ng, T.-T. and Chang, S.-F. (2006). An online system for classifying computer graphics images from natural photographs. In *SPIE Electronic Imaging*, San Jose, CA.
- [69] Ng, T.-T., Chang, S.-F., Hsu, Y.-F., and Pepeljugoski, M. (2005a). Columbia

- photographic images and photorealistic computer graphics dataset. ADVENT Technical Report 205-2004-5, Columbia University.
- [70] Ng, T.-T., Chang, S.-F., Hsu, Y.-F., Xie, L., and Tsui, M.-P. (2005b). Physics-motivated features for distinguishing photographic images and computer graphics. In *ACM Multimedia*, Singapore.
- [71] Ng, T.-T., Chang, S.-F., and Sun, Q. (2004). Blind detection of photomontage using higher order statistics. In *IEEE International Symposium on Circuits and Systems*, Vancouver, Canada.
- [72] Nicodemus, F. E., Richmond, J. C., Hsia, J. J., Ginsberg, I., and Limperis, T. (1977). Geometric considerations and nomenclature for reflectance. Monograph 160, National Bureau of Standards (US).
- [73] Nikias, C. and Petropulu, A. (1993). *Higher-order spectra analysis: a nonlinear signal processing framework*. PTR Prentice Hall.
- [74] Nishino, K. and Nayar, S. (2004). The world in an eye. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- [75] Patwardhan, K. A., Sapiro, G., and Bertalmio, M. (2005). Video inpainting of occluding and occluded objects. In *IEEE International Conference on Image Processing*.
- [76] Pentland, A. (1985). On describing complex surface shapes. *Image and Vision Computing*, 3(4):153–162.
- [77] Platt, J. C. (1999). *Advances in Large Margin Classifiers*, chapter Probabilities for Support Vector Machines, pages 61–74. MIT Press.

- [78] Popescu, A. and Farid, H. (2004a). Exposing digital forgeries by detecting duplicated image regions. Technical Report TR2004-515, Computer Science, Dartmouth College.
- [79] Popescu, A. and Farid, H. (2004b). Statistical tools for digital forensics. In *6th International Workshop on Information Hiding*, Toronto, Canada.
- [80] Popescu, A. and Farid, H. (2005a). Exposing digital forgeries by detecting traces of re-sampling. *IEEE Transactions on Signal Processing*, 52(2):758–767.
- [81] Popescu, A. and Farid, H. (2005b). Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing*, 53(10):3948–3959.
- [82] Ranck, R. (1990). The camera does lie. *The New York Times*.
- [83] Rosales, R., Achan, K., and Frey, B. (2003). Unsupervised image translation. In *IEEE International Conference on Computer Vision*, pages 472–478.
- [84] Rudin, L., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1-4):259–268.
- [85] Schneider, M. and Chang, S.-F. (1996). A robust content based digital signature for image authentication. In *IEEE International Conference on Image Processing*, volume 3.
- [86] Shum, H.-Y. and Kang, S. B. (2000). A review of image-based rendering techniques. In *IEEE/SPIE Visual Communications and Image Processing*, Perth, Australia.
- [87] Sigl, J. and Chamoun, N. (1994). An introduction to bispectral analysis for the electroencephalogram. *Journal of Clinical Monitoring and Computing*, 10(6):392–404.

- [88] Simoncelli, E. P. (1999). Modelling the joint statistics of images in the wavelet domain. In *SPIE 44th Annual Meeting*, Denver, CO.
- [89] Smith, J. R. and Chang, S.-F. (1997). Visually searching the web for content. *IEEE Multimedia*, 4(3):12–20.
- [90] Snavely, N., Seitz, S. M., and Szeliski, R. (2006). Photo tourism: Exploring photo collections in 3D. In *ACM SIGGRAPH*.
- [91] Sochen, N., Kimmel, R., and Malladi, R. (1998). A general framework for low level vision. *IEEE Transactions on Image Processing*, 7(3):310–318.
- [92] Srivastava, A., Lee, A. B., Simoncelli, E. P., and Zhu, S.-C. (2003). On advances in statistical modeling of natural images. *Journal of Mathematical Imaging and Vision*, 18(1):17–33.
- [93] Swaminathan, A., Wu, M., and Liu, K. J. R. (2006). Image tampering identification using blind deconvolution. In *IEEE International Conference on Image Processing*, pages 2311–2314, Atlanta, GA.
- [94] Tewfik, A. and Mansour, M. (2002). Secure watermark detection with non-parametric decision boundaries. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*.
- [95] Tseng, B. L., Lin, C.-Y., Naphade, M., Natsev, A., and Smith, J. R. (2003). Normalized classifier fusion for semantic visual concept detection. In *IEEE International Conference in Image Processing*, pages 535–538.
- [96] Tsin, Y., Ramesh, V., and Kanade, T. (2001). Statistical calibration of CCD imaging process. In *IEEE International Conference on Computer Vision*.



- [97] Venturini, I. (2004). Counteracting oracle attacks. In *ACM multimedia and security workshop on Multimedia and security*, pages 187–192, Magdeburg, Germany.
- [98] Vese, L. and Osher, S. (2003). Modeling textures with total variation minimization and oscillating patterns in image processing. *Journal of Scientific Computing*, 19(1):553–572.
- [99] Vieville, T. and Faugeras, O. D. (1992). Robust and fast computation of unbiased intensity derivatives in images. In *European Conference on Computer Vision*, pages 203–211, Santa Margherita Ligure, Italy.
- [100] Wang, W. and Farid, H. (2006). Exposing digital forgeries in video by detecting double MPEG compression. In *ACM Multimedia and Security Workshop*, Geneva, Switzerland.
- [101] Wang, Y. and Moulin, P. (2006). On discrimination between photorealistic and photographic images. In *IEEE International Conference on Acoustics, Speech and Signal Processing*.
- [102] Weiss, I. (1993). Noise-resistant invariants of curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):943–948.
- [103] Wong, P. W. (1998). A watermark for image integrity and ownership verification. In *Proc. IS&T Conference on Image Processing, Image Quality and Image Capture Systems*, pages 128–140. Portland, Oregon.
- [104] Wu, M. and Liu, B. (1998). Watermarking for image authentication. In *IEEE International Conference on Image Processing*.

- [105] Xiao, F., Farrell, J., DiCarlo, J., and Wandell, B. (2003). Preferred color spaces for white balancing. In *SPIE Electronic Imaging Conference*, volume 5017.
- [106] Yeung, M. M. and Mintzer, F. (1997). An invisible watermarking technique for image verification. In *IEEE International Conference on Image Processing*, pages 680–683.